# Appendices

(last updated on May 8, 2020)

# Contents

# Appendix A

# Details of the Multidisciplinary Exam of the BS degree

The purpose of the Multidisciplinary Exam exam is to ensure that the subjects are studied and understood cohesively in a perspective of a global and comprehensive education process, and not just individually and in insulation.

It is administered after the termination of the second year of studies, during the Fall semester of the third year, typically at the end of the lectures.

The exam covers the content of the following (group of) courses:

1. *Introduction to Programming* I and II
2. *Data Structures and Algorithms* I and II
3. *Discrete Math*
4. *Mathematical Analysis* I and II
5. *Differential Equations*
6. *Computer Architecture*
7. *Analytic Geometry and Linear Algebra* I and II
8. *Probability and Statistics*
9. *Data Modeling and Databases* I and II
10. *Operating Systems*
11. *Introduction to AI*
12. *Information Theory*
13. *Programming Paradigms*

14. *Networks*
15. *Theoretical Computer Science*

For every course or group of courses, the dean via the associate dean for teaching proposes to the department of education one or more reference textbook(s) or a part of them, after consultation with the current instructor of the course. Such text is used as the basis for defining the questions.

A suitable committee proposed by the dean via the associate dean for teaching and approved by the department of education defines, oversees, and grades the exam.

The exam:

1. consists of 30 questions, 2 for each course or group of courses mentioned above,
2. is closed book,
3. lasts 4 hours,
4. is graded:
   (a) considering the minimal level for sufficiency to have:
      i. answered correctly at least 16 questions, and
      ii. answered correctly at least one question in 10 different courses or group of courses,
   (b) assigning:
      i. an “A” to the students with at least 25 correct answers, and at least one answer per course or group of courses,
      ii. a “B” to the students below the level of A, with at least 19 correct answers, and at least one answer in 13 different courses or group of courses,
      iii. a “C” to the students passing the exam for the rules above but not achieving the level of the B,
5. has the first retake after the winter break,
6. has a second retake at end of the Spring semester of the third year.

# Appendix B

# Details of the Elaboration and the Evaluation of the Thesis of the BS

## B.1 Introduction

A BS thesis is an essential part of the learning experience for students as it focuses on their ability to integrate knowledge acquired throughout courses into a concrete problem. The final thesis assessment will consist of two parts. The first part relates to the project outcome. The second part relates to the project documentation.

The part that relates to the project outcome can take two forms, depending on the nature of the goals of the project: system development (`thesis type 1`) or research-oriented investigation (`thesis type 2`).

In the case of a thesis that focuses on system development, the outcome will take the form of the design, implementation, test, and evaluation of a complete system.

In the case of a thesis that focuses on a research-oriented investigation, the outcome will take the form of an analysis of the subject area, a synthesis of a specific conjecture or hypothesis, empirical validation of the proposed model, and critical appraisal of the empirical or theoretical results.

The part that relates to the project report, i.e. the thesis itself, is assessed under the headings of

- organisation and clarity,
- technical contents,

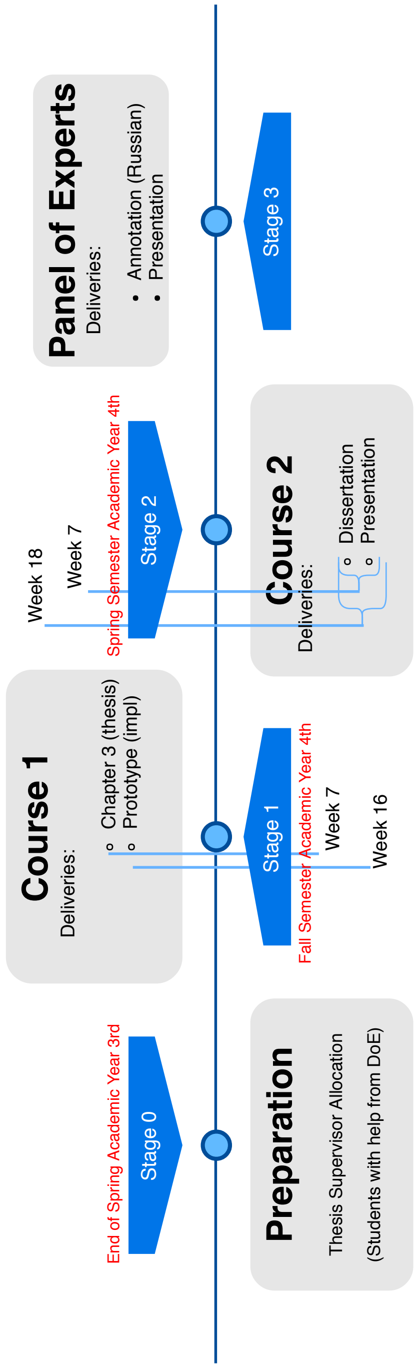
- conclusions and references.

This document serves as guidelines to ensure the maximum quality of students' work and the possibility of them finishing their thesis in time. The thesis work will be developed in an incremental manner: students will have a running prototype of the thesis by the end of the first academic term, and a complete development by the end of the academic year.

Lastly, it is important to emphasize that during the thesis, students are expected to meet regularly with their supervisor and to act diligently on them.

This document is organised as follows. Chapter B.2 depicts a general time line for students to follow. The specifics of each phase are presented in the rest of the chapters. Chapter B.3 describes what needs to be done before starting the thesis work. Chapters B.4 and B.5 give guidelines for the thesis document and the presentation, respectively. Finally, chapter B.6 presents the evaluation criteria for the thesis work.

## B.2 Thesis work: time line



## B.3 Preparation

At the end of the third year Fall semester, students need to have a meeting with a potential supervisor (IU Faculty) to decide on the topic of the thesis, research questions, title, aims, objectives, and the general structure of the thesis project. This activity is carried out by students with the help of the Department of Education.

Students, with their potential supervisors' help, should provide a complete specification of the problem, listing objectives, requirements and a work plan to develop. The specification should include the research question.

- For `thesis type 1`, students should provide a description of the functionality of the system to be designed. Along with a discussion of the parameters affecting the performance of the system, its limitations and restrictions.
- For `thesis type 2`, students should provide a description of the goals of the investigation to be carried out in the project. Along with a discussion of the criteria or metrics to be used to assess the outcome of the investigation.

## B.4 Dissertation

The BS dissertation is an independently written scientific work. This chapter shows the general guidelines for the document (each section can be treated as a chapter of the dissertation).

### B.4.1 Title, abstract, acknowledgements, table of contents

Thesis' cover can be found in the thesis template: `https://www.overleaf.com/read/tgfbfnjjkkyx`template.

### B.4.2 Introduction

In this chapter, students should introduce their work. Students should

- mention what the topic is about;
- why there is a need to further research on this topic;

- what the hypotheses are;
- state the research questions.

### B.4.3 Literature Review and Related Work

In this chapter, students should write a Literature Review and Related Work on the topics and domains related to the project. For this, students may consult books, research journals, conference proceedings, handbooks, and even online courses. Some of the questions to be covered in this chapter are

- What is the relevant prior work?
- Where can I find it? (citations are important)
- Why should it be done differently?
- Has anyone attempted your approach previously?
- Where is that work reported?
- What is the outline of your way.

### B.4.4 Design and Methodology

In this chapter, students should describe the general structure of the study: methods, approaches or processes followed during the study. The methodology should be linked back to the aims of the thesis and the literature to explain why you are using certain methods. In the case of a system development, students should describe the Design of the system: describe each component and how they will interact to each other. They should also describe the different experiments carried out. The design and solution should be elegant and robust – consistent with industrial standards.

This chapter should answer:

- Why did you use this technology/method?
- How does the theory relate to your implementation?
- What are your underlying assumptions?
- What did you neglect and what simplifications have you made?
- What tools and methods did you use?
- Why use these tools and methods?

### B.4.5 Implementation and Results

In this chapter, students should describe in detail the implementation performed, as well as the results of applying the system implementation to a case study or any experiment undertaken.

- For `thesis type 1`: Test suites implemented and documented. Faults also documented and catalogued. Comparison of the system performance to those other systems described in Section B.4.3 or with benchmark metrics.
- For `thesis type 2`: Results presented in a manner that makes explicit their relationship to the research question. Including some estimate of error or reliability, and a comparison to those other systems described in Section B.4.3 or with benchmark metrics.

This chapter should answer:

- Did you actually build it?
- How can you test it?
- How did you test it?
- Why did you test it this way?

### B.4.6 Analysis and Discussion

In this chapter, the student should describe in detail the analysis and discussion of the findings from previous chapters. Students should discuss the accuracy and relevance of the results; compare with other researchers results.

Are the results satisfactory? how do you define success in your thesis work. Why should you (not) test it more? What compensations had to be made to interpret the results? Why did you succeed/fail?

### B.4.7 Conclusion

In this chapter, students should state the consequences of the achieved results. They should also mention whether the results are satisfactory and how they can be improve.

This chapter is a short account of the results of your work, emphasising mainly what is new.

Students should align conclusion to the Introduction, in which the problem was described. They should also mention limitations of the work and suggest what further work might be done. Make a synthesis of the contributions and impact of your work and recommendations.

### B.4.8 Bibliography

References should be consistently cited in the text. (Plagiarism is unacceptable). References of the dissertation should follow APA (American Psychological Association) style (it is already provided in the latex template). References to the World Wide Web, Wikipedia, non peer-reviewed sites, non-academic manuscripts and other blogs/social networks **should be avoided**.

Some of the questions this chapter should answer are:

- What is the background reading list?
- Where is the related work?
- Where is the prior work?
- Where can I find important material?

## B.5 Presentation

Students, in addition to submitting the thesis document, need to make a presentation to their supervisors describing the thesis work. Supervisors arrange the presentation's schedule and location. They might invite other IU Faculty to attend the presentation.

Students will have 20 minutes only for the presentation. It will be followed by 10 minutes of questions/answers. The presentation should focus on the problem of the thesis, the research questions, the work performed, what was discovered, what are the lessons learned and the recommendations.

This is the recommended structure of the presentation

- General context
- Open problem you addressed – research questions

- literature review and background
- Your proposed solution to the problem – design and methodologies
- implementation and experiments
- results and discussion
- conclusions – contribution, impact, future work

## B.6 Evaluation Criteria

The thesis work is an activity to be developed by students during the final academic year of the Bachelor program. The work is worth 2 courses (one in Fall and one in Spring). It is evaluated is several stages. The following sections describe the different stages of the thesis evaluation, the grading criteria and the committee in charge of the evaluation.

### B.6.1 Stage 1: Course 1 (in Fall)

There are 2 evaluations during the Fall semester:

1. Mid of semester assessment is based on the Literature Review and Related Work thesis chapter. The writing is aligned with the Academic Writing and Research Culture course. This grade represents the projected grade at the end of the semester of the student if the performances remain those of the initial part of the semester.

   **Deadline:** Week 7 in the Fall semester.
   **Who:** Supervisor
   **Where:** Moodle (A/B/C/D)

2. Final evaluation of the course is based on the Literature Review and Related Work thesis chapter and a working prototype of the thesis. E.g. For `thesis type 1`, the student should present a working prototype of the implementation.

   **Deadline:** Week 17 in the Fall semester.
   **Who:** Supervisor
   **Where:** Moodle (A/B/C/D)

### B.6.2 Stage 2: Course 2 (in Spring)

There are 2 evaluations during the Spring semester:

1. Mid of semester assessment is based on Chapters Design and Methodology and Implementation chapters. This grade represents the projected grade at the end of the semester of the student if the performances remain those of the initial part of the semester.

   **Deadline:** Week 7 in the Spring semester.
   **Who:** Supervisor
   **Where:** Moodle (A/B/C/D)

2. Final evaluation of the course is based on the thesis work and Presentation (detailed criteria below).

   **Deadline:** Week 18 in the Spring semester.
   **Who:** Supervisor
   **Where:** Moodle (A/B/C/D)

Submission of the final manuscript of the thesis is via Moodle. Students should provide a hard copy of the thesis after Supervisor's approval (between Week 16 and 18 of the Spring semester)

The grading form for the Thesis can be found here: `https://drive.google.com/open?id=1OZC7OOXKNE-yGXwP803fXE7k-Ugy1mWe`Final Thesis Assessment.

These are the criteria:

For `thesis type 1`:

- Does the student show that the design meets the requirements & specification?
- Has a theoretical model been used or developed?
- Has a formal design methodology been adopted and, if so, properly used?
- Is the design well structured (e.g. is it hierarchical, are there well-identified interfaces and is there a functional specification for each sub-system)?
- Is the design of the software or hardware fully implemented?
- Is the implementation modular?
- Is the final system of high quality (i.e. elegant or robust implementation consistent with industrial standards)?

For `thesis type 2`

- Are all the factors, empirical or theoretical, affecting the investigation analysed and coherently summarized?
- Has a theoretical model been used or developed?
- Is a clear research question in the form of an explicit conjecture or hypothesis formulated and discussed?
- Are appropriate validation criteria identified and described?
- Have appropriate studies been conducted sufficiently well to address the research question (by simulation or other means)?
- Are the empirical tests well designed with clearly defined parameters and measurable outcomes?
- Are the empirical tests sufficient to arrive at a convincing conclusion or answer to the research question?

**For the thesis document:** Organization & Clarity

- Is the abstract representative of the content of the report?
- Is there evidence that the student has read a representative amount of relevant material?
- Is this material properly cited?
- Is the report logical in its development of the material?
- Are grammar and spelling generally correct?
- Is the phraseology clear and concise?

Technical Content

- Does the student demonstrate mastery of the domain?
- Is the student's synthesis of the relevant material compelling?
- Has the student explained all the issues in his or her own words or recycled significant amount of the text from other sources?

Conclusions and Future Work

- Are conclusions well written?
- Are the student's conclusions or insights significant?

- Has he or she presented an objective and thorough appraisal of his or her achievements?
- Does the student make clear what he or she has learned from the project?
- Does he or she present ideas on future work related to his or her project?
- Does he or she understand the relevance and importance of the work presented in the Thesis?

**For the Presentation**

- Has the student covered all of the relevant issues?
- Was the presentation clear and concise?
- Was the student confident of his or her subject matter?
- Did he or she make the subject matter interesting?
- Did he or she answer questions well?

### B.6.3 Stage 3: Panel of Experts

...

Students need to provide an *Annotation* in Russian. *Annotation* (Extended Abstract) is a summary of the Thesis in Russian (10-15 pages) with the same structure, executed in the same template, with the only change in the document title: `Thesis` replaced with *Annotation*. The deadline for this document is Week 16 of Spring semester. Students also need to provide a presentation of the thesis in front of a Russian Committee.

...

### B.6.4 Outstanding Thesis Works

Those outstanding thesis works, works published or accepted in A or B venues in the Faculty ranking, will receive a Dean's List recognition.

# Appendix C

# General Key Competences Presented in the First Two Years of Instruction of the BS degree

1. Algorithm analysis
2. Algorithm strategies
3. Artificial intelligence
4. Assembly language machine organization
5. Assembly language representation of data
6. Automata
7. Big data
8. Calculus
9. Code generation
10. Compiler design and construction
11. Compiler semantics analysis
12. Computational complexity
13. Computational logic
14. Concurrency
15. Data modelling
16. Data structures and algorithms
17. Database systems and design
18. Digital logic and digital systems
19. Differentiation
20. Distributed systems and computing
21. Evolutionary algorithms
22. Fault tolerance
23. File systems
24. Formal methods
25. Formal models and semantics
26. Functional programming
27. Generic programming
28. indexing
29. Information retrieval and storage
30. Integration
31. Internet applications
32. Language pragmatics

33. Linear algebra
34. Linear systems
35. Local area networks
36. Machine learning
37. Machine level representation of data
38. Memory management
39. Memory organization and architectures
40. Mobile platforms
41. Modeling and simulation
42. Multiprocessing and alternative architectures
43. Network security
44. Networked applications
45. Numerical analysis
46. Numerical optimization
47. Object oriented programming
48. Operating systems
49. Parallel algorithms, analysis and programming
50. Parallel architectures
51. Parallel performance
52. Parallel fundamentals
53. Performance enhancements
54. Platform security
55. Probability
56. Programming
57. Programming constructs
58. Proof techniques
59. Query languages
60. Real time and embedded systems
61. Relational databases
62. Requirements engineering
63. Resource allocation and scheduling
64. Run-time systems
65. Scheduling and dispatch
66. Sequences
67. Sets, relations, and functions
68. Software design
69. Software engineering
70. Software evolution
71. software reliability
72. Software verification and validation
73. State and state machines
74. Static analysis
75. System performance evaluation
76. System security
77. Virtual machines
78. Web platforms
79. Web security
80. Web services

# Appendix D

# Detailed Breakdown of Knowledge Areas in the last two years of the streams of the BS degree

Tables I.1, I.3, and I.2 contain the breakdown of the knowledge areas covered in each core course taught in the last two years of the three streams, the Software Engineering, Data Science, and Robotics streams, respectively.

**Table D.1:** Breakdown of the knowledge areas covered in each core course of instruction in 2 years of the Software Engineering Stream– the numbers in the cells are percentages.

| Courses | | IS | GV | PL | PD | CE | OS | CN | Ph | FN | AL | HCI | SP | CM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Information Theory* | CSE | 80 | 20 | | | | | | | | | | | |
| *System Theory* | CSE | | | | | 100 | | | | | | | | |
| *Compilers Construction* | CSE | | | 50 | | | | | | | 50 | | | |
| *Cloud Computing* | CSE | | | | 100 | | | | | | | | | |
| *Introduction to Big Data* | CSE | 70 | | | | | | | | | 30 | | | |
| *Philosophy* 2 (Languages and Perceptions) | H | | | | | | | | 100 | | | | | |
| *Introduction to Machine Learning* | CSE | 100 | | | | | | | | | | | | |
| *Digital Signal Processing* | CSE | | | | | | | | | | | | | 100 |
| *Information Retrieval* | CSE | 100 | | | | | | | | | | | | |
| *Data Mining* | CSE | 100 | | | | | | | | | | | | |
| *Embedded and Shared Memory Programming* | CSE | | | 80 | | | 20 | | | | | | | |
| *Numerical Modeling* | CSE | | | | | | | 100 | | | | | | |
| *Business Analytics* | CSE | 70 | | | | | | | | 30 | | | | |
| *Academic Research and Writing Culture* | H | | | | | | | | | | | | 100 | |
| *Distributed and Network Programming* | CSE | | | | 100 | | | | | | | | | |
| *Advanced Databases* | CSE | | | | | | | 50 | | | 50 | | | |
| *Advanced Algorithms* | CSE | | | | | | | | | | | | | |
| *Human Computer Interfaces* | CSE | | | | | | | | | | | 100 | | |
| *Life Safety* | H | | | | | | | | | | | | 100 | |

**Table D.3:** Breakdown of the knowledge areas covered in each core course of instruction in 2 years of the Data Science Stream– the numbers in the cells are percentages.

| Courses | | IS | GV | PL | PD | CE | OS | CN | Ph | FN | AL | HCI | SP | CM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Information Theory* | CSE | 80 | 20 | | | | | | | | | | | |
| *System Theory* | CSE | | | | | 100 | | | | | | | | |
| *Compilers Construction* | CSE | | | 50 | | | | | | | 50 | | | |
| *Cloud Computing* | CSE | | | | 100 | | | | | | | | | |
| *Introduction to Big Data* | CSE | 70 | | | | | | | | | 30 | | | |
| *Philosophy* 2 (Languages and Perceptions) | H | | | | | | | | 100 | | | | | |
| *Introduction to Machine Learning* | CSE | 100 | | | | | | | | | | | | |
| *Digital Signal Processing* | CSE | | | | | | | | | | | | | 100 |
| *Information Retrieval* | CSE | 100 | | | | | | | | | | | | |
| *Data Mining* | CSE | 100 | | | | | | | | | | | | |
| *Embedded and Shared Memory Programming* | CSE | | | 80 | | | 20 | | | | | | | |
| *Numerical Modeling* | CSE | | | | | | | 100 | | | | | | |
| *Business Analytics* | CSE | 70 | | | | | | | | 30 | | | | |
| *Academic Research and Writing Culture* | H | | | | | | | | | | | | 100 | |
| *Distributed and Network Programming* | CSE | | | | 100 | | | | | | | | | |
| *Advanced Databases* | CSE | | | | | | | 50 | | | 50 | | | |
| *Advanced Algorithms* | CSE | | | | | | | | | | | | | |
| *Human Computer Interfaces* | CSE | | | | | | | | | | | 100 | | |
| *Life Safety* | H | | | | | | | | | | | | 100 | |

**Table D.5:** Breakdown of the knowledge areas covered in each core course of instruction in 2 years of the Robotics Stream– the numbers in the cells are percentages.

| Courses | | IS | GV | PL | M | P | EE | CE | OS | MP | Ph | ME | BE | SP | CM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Information Theory* | CSE | 80 | 20 | | | | | | | | | | | | |
| *System Theory* | CSE | | | | | | | 100 | | | | | | | |
| *Mathematical Modeling* | M | | | | 100 | | | | | | | | | | |
| *Theoretical Mechanics* | M | | | | | | | | | 100 | | | | | |
| *Introduction to Robotics* | CSE | 100 | | | | | | | | | | | | | |
| *Philosophy* 2 (Languages and Perceptions) | H | | | | | | | | | | 100 | | | | |
| *Introduction to Machine Learning* | CSE | 100 | | | | | | | | | | | | | |
| *Digital Signal Processing* | CSE | | | | | | | | | | | | | | 100 |
| *Control Theory* | CSE | | | | | | | 100 | | | | | | | |
| *Mechanics and Machines* | P | | | | | 100 | | | | | | | | | |
| *Embedded and Shared Memory Programming* | CSE | | | 80 | | | | | 20 | | | | | | |
| *Mechatronics* | CSE | | | | | | 50 | | | | | 50 | | | |
| *Introduction to Computer Vision* | CSE | 100 | | | | | | | | | | | | | |
| *Academic Research and Writing Culture* | H | | | | | | | | | | | | | 100 | |
| Nonlinear Control Theory | CE | | | | | | | 100 | | | | | | | |
| *Sensors and Sensing* | CSE | 100 | | | | | | | | | | | | | |
| Fundamentals of graphics and virtual realities | CSE | | 100 | | | | | | | | | | | | |
| *Microcontrollers and Embedded Hardware* | CSE | | | | | | | | | | | | 100 | | |
| *Life Safety* | H | | | | | | | | | | | | | 100 | |

# Appendix E

# Languages, Technologies, and Tools used in the Courses of the BS Degree

In the context of the course, several (programming) languages, technologies, and tools are used. Here below there is a list of some of the most relevant.

## E.1 (Programming) Languages

- bash scripting language
- C
- C++
- Eiffel
- Java
- LISP
- MIPS assembly language
- OCL
- Python
- SQL
- UML

## E.2 Tools and Technologies

- Anaconda (https://anaconda.org/)
- Apache Hadoop (http://hadoop.apache.org/) with MapReduce (https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html)
- CLion (https://www.jetbrains.com/clion/)
- Docker (https://www.docker.com/)
- Eclipse (http://www.eclipse.org/)
- Intellij IDEA (https://www.jetbrains.com/idea/)
- Keras (https://keras.io/)
- Matlab / Simulink (https://www.mathworks.com/products/simulink.html)
- Networkx (https://networkx.github.io/)
- NoSQL: Graph DB (https://neo4j.com/developer/)
- Numpy (http://www.numpy.org/)
- Octave (https://www.gnu.org/software/octave/)
- Orange (https://orange.biolab.si/)
- Pandas (https://pandas.pydata.org/)
- Pomegranade (http://www.pomegranateapps.com/mac/)
- PyCharm (https://www.jetbrains.com/pycharm/)
- Scikit-learn (http://scikit-learn.org/)
- Scilab/Xcos (https://www.scilab.org/en)
- WEKA (https://www.cs.waikato.ac.nz/ml/weka/)
- Tensorflow (https://www.tensorflow.org/)
- Wireshark (https://www.wireshark.org/)
- *nix command-line (traceroute, ping, ssh, flex, bison ...)

In addition, the students experiment the Windows and Linux operating system families, in their most common versions.

# Appendix F

# Specific Courses for the Direction in Finance Technologies of the BS Degree

The following are courses in the direction of finance technologies of the stream of data science that substitute for the three electives.

- Industrial Programming in Java (described in Section L.19 on page 552)
- Industrial Software Testing (described in Section L.20 on page 553)
- Introduction to Communication (described in Section L.25 on page 554)

# Appendix G

# Details of the Elaboration and the Evaluation of the Thesis for the MS Degree

## G.1 Introduction

A MS thesis is an essential part of the learning experience for students as it focuses on their ability to integrate knowledge acquired throughout courses into a concrete problem.

The final thesis assessment will consist of two parts. The first part relates to the project outcome. The second part relates to the project documentation.

The part that relates to the project outcome can take two forms, depending on the nature of the goals of the project: system development (`thesis type 1`) or research-oriented investigation (`thesis type 2`).

In the case of a thesis that focuses on system development, the outcome will take the form of the design, implementation, test, and evaluation of a complete system.

In the case of a thesis that focuses on a research-oriented investigation, the outcome will take the form of an analysis of the subject area, a synthesis of a specific conjecture or hypothesis, empirical validation of the proposed model, and critical appraisal of the empirical or theoretical results.

The part that relates to the project report, i.e. the thesis itself, is assessed under the headings of

- organisation and clarity,
- technical contents,
- conclusions and references.

This document serves as guidelines to ensure the maximum quality of students' work and the possibility of them finishing their thesis in time. The thesis work will be developed in an incremental manner: students will have a running prototype of the thesis by the end of the first academic term, and a complete development by the end of the academic year.

Each section of this document defines a phase for the development of the thesis and lists specific outcomes and, when applicable, evaluation criteria. Deadlines are defined for each phase.

Lastly, it is important to emphasize that during the thesis, students are expected to meet regularly with their supervisor and to act diligently on them.

## G.2 Thesis approval

To guarantee the overall compliance to the program objectives, each thesis proposal needs to be approved by the director of the institute to which the specific program belongs.

## G.3 Specification of the problem to solve

Students, with their supervisor's help, should provide a complete specification of the problem, listing objectives, requirements and a work plan to develop. The specification should include the research question.

- For `thesis type 1`, students should provide a description of the functionality of the system to be designed. Along with a discussion of the parameters affecting the performance of the system, its limitations and restrictions.
- For `thesis type 2`, students should provide a description of the goals of the investigation to be carried out in the project. Along with a discussion of the criteria or metrics to be used to assess the outcome of the investigation.

### G.3.1 Deadline

Week 2 (approx. August 31)

### G.3.2 Outcome

- ⋆ Initial review of the system specification. This will be part of the Introduction for the thesis document.

### G.3.3 Criteria for evaluation

Defined by supervisors based on the clarity and completeness of the project objectives, requirements, and plan.

## G.4 Literature Review

Students should produce a Literature Review on the topics and domains related to the project. For this, students may consult books, research journals, conference proceedings, handbooks, and even online courses.

### G.4.1 Deadline

4 Week (approx. September 15)

### G.4.2 Outcome

- ⋆ An initial version of the Literature Review chapter for the thesis (Ch. 2).

### G.4.3 Criteria for evaluation

Defined by supervisors.

### G.4.4 Mid of first semester evaluation

Mid of first semester assessment from the supervisor. Deadline: September 15 (hard deadline).

**Criteria for evaluation** Mid of semester assessment (A/B/C/D) on Moodle based on previous phases, representing the projected grade at the end of the semester of the student if the performances remain those of the initial part of the semester.

## G.5 Methodology and Prototype

**Methodology:** Students should write the description of the method, approach or process that was implemented. Students should do this in the light of the aims and objectives of the project, and the literature review performed in previous phases.

**Prototype:** Students should have a working implementation of the solution:

- `thesis type 1`: prototype of the design of the software. A working implementation of the system that enables students to get results.
- `thesis type 2`: studies conducted addressing the research question.

### G.5.1 Deadline

Week 11 (approx. November 1.)

### G.5.2 Outcome

- ⋆ Prototype of the system.
- ⋆ Methodology chapter for the thesis (Ch. 3)
- ⋆ Implementation chapter for the thesis (Ch. 4)

### G.5.3 Criteria for evaluation

Defined by supervisors.

## G.6 Evaluation and Discussion

Students should implement the evaluation (validation) of the system and write the discussion of the findings.

- `thesis type 1`: Test suites already implemented and documented. Faults also documented and catalogued. Comparison of the system performance to those other systems described in Section G.4 or with benchmark metrics.
- `thesis type 2`: Results presented in a manner that makes explicit their relationship to the research question. Including some estimate of error or reliability, and a comparison to those other systems described in Section G.4 or with benchmark metrics.

### G.6.1 Deadline

Week 16

### G.6.2 Outcome

⋆ Evaluation and Discussion chapter for the thesis (Ch. 5)

### G.6.3 Criteria for evaluation

Defined by supervisors.

### G.6.4 Formal grade on Moodle

Final evaluation from the supervisor: first part of the academic year. Deadline December 10 (hard deadline)

**Criteria for evaluation** Formal grade on Moodle by the supervisor

- A (All is perfect)
- B (requires aesthetic improvements)
- C (requires some content improvement)
- D (Incomplete work) – Time till January to complete

For those students who failed, they should go back to previous chapters to change aims and objectives of the project, reflect new methodology and the associated literature review, provide details of the new implementation and so on. Final Deadline for failing students is January 15 (hard deadline)

**Criteria for evaluation** Formal grade on Moodle by the supervisor

- B (All is perfect)
- C (requires aesthetic improvements)
- D (fail)

## G.7 Paper Submission

Students should write a scientific paper based on the findings and previous thesis chapters.

Students need to improve the paper writing. Then submit the paper to a scientific venue proposed by the supervisor, or the supervisor can suggest a committee of two Faculty members of IU. In this case, the paper should not exceed 4 pages.

### G.7.1 Deadline

In both cases, the deadline for acceptance must not exceed the final day of classes.

### G.7.2 Outcome

⋆ A scientific paper.

### G.7.3 Criteria for evaluation

External evaluation.

## G.8 Improvement of the Literature Review

Students should provide a systematic Literature Review according to the new findings.

### G.8.1 Deadline

Week 24

### G.8.2 Outcome

- ⋆ Final version of the Literature Review thesis chapter (Ch. 2)

### G.8.3 Criteria for evaluation

Defined by supervisors.

### G.8.4 Mid of first semester evaluation

Mid of first semester assessment from the supervisor. Deadline: February 15 (hard deadline).

**Criteria for evaluation** Mid of semester assessment (A/B/C/D) on Moodle based on previous phases, representing the projected grade at the end of the semester of the student if the performances remain those of the initial part of the semester.

## G.9 Improvement of the Implementation and System design

Students should provide a final system design with a final solution implementation. The design and solution should be elegant and robust – consistent with industrial standards.

### G.9.1 Deadline

Week 28

### G.9.2 Outcome

- ⋆ Final implementation of the system.
- ⋆ Final version of the Methodology chapter for the thesis (Ch. 3)
- ⋆ Final version of the Implementation chapter for the thesis (Ch. 4).

### G.9.3 Criteria for evaluation

Defined by supervisors based on the following questions.

For `thesis type 1`:

- Does the student show that the design meets the requirements & specification?
- Has a theoretical model been used or developed?
- Has a formal design methodology been adopted and, if so, properly used?
- Is the design well structured (e.g. is it hierarchical, are there well-identified interfaces and is there a functional specification for each sub-system)?
- Is the design of the software or hardware fully implemented?
- Is the implementation modular?
- Is the final system of high quality (i.e. elegant or robust implementation consistent with industrial standards)?

For `thesis type 2`

- Are all the factors, empirical or theoretical, affecting the investigation analysed and coherently summarized?
- Has a theoretical model been used or developed?
- Is a clear research question in the form of an explicit conjecture or hypothesis formulated and discussed?
- Are appropriate validation criteria identified and described?
- Have appropriate studies been conducted sufficiently well to address the research question (by simulation or other means)?
- Are the empirical tests well designed with clearly defined parameters and measurable outcomes?
- Are the empirical tests sufficient to arrive at a convincing conclusion or answer to the research question?

## G.10 Improvement of the Evaluation and Discussion

Students should repeat phase described in Section G.6. Then report new findings (if any).

### G.10.1 Deadline

Week 30

### G.10.2 Outcome

⋆ Final version of the Evaluation and Discussion chapter for the thesis (Ch. 5).

### G.10.3 Criteria for evaluation

Defined by supervisors based on the following questions.

For `thesis type 1`:

- Are the test cases clearly identified and test results clearly documented?
- Are all faults clearly documented and catalogued? Has the student compared the system performance with those of other systems or with benchmark metrics?

Thesis type 2:

- Have the results been clearly presented in a manner that makes explicit their relationship to the research question?
- Do the results include some estimate of error or reliability? Has the student compared the results with those of other systems or with benchmark metrics?

## G.11 Thesis document

Students should write up the Abstract, Introduction and Conclusions and Future Work of the thesis document. They should also put together all chapters, then check the document to ensure consistency in style and content, and to improve thesis readability. Thesis template can be found `https://www.overleaf.com/read/tgfbfnjjkkyx`here.

They should also prepare a final demo and presentation for the supervisor.

### G.11.1 Deadline

Week 34

### G.11.2 Outcome

- ⋆ Final Thesis document
- ⋆ Demo
- ⋆ Presentation

### G.11.3 Criteria for evaluation

Defined by supervisors based on the following questions.

**For the thesis document:** Organization & Clarity

- Is the abstract representative of the content of the report?
- Is there evidence that the student has read a representative amount of relevant material (a minimum 10 references are required)?
- Is this material properly cited?
- Is the report logical in its development of the material?
- Are grammar and spelling generally correct?
- Is the phraseology clear and concise?

Technical Content

- Does the student demonstrate mastery of the domain?
- Is the student's synthesis of the relevant material compelling?
- Has the student explained all the issues in his or her own words or recycled significant amount of the text from other sources?

Conclusions and Future Work

- Are conclusions well written?
- Are the student's conclusions or insights significant?
- Has he or she presented an objective and thorough appraisal of his or her achievements?
- Does the student make clear what he or she has learned from the project?
- Does he or she present ideas on future work related to his or her project?
- Does he or she understand the relevance and importance of the work presented in the Thesis?

**For the Presentation**

- Has the student covered all of the relevant issues?
- Was the presentation clear and concise?
- Was the student confident of his or her subject matter?
- Did he or she make the subject matter interesting?
- Did he or she answer questions well?

### G.11.4 Formal grade on Moodle

Final evaluation from the supervisor: second part of the academic year. Deadline April 30 (hard deadline)

**Criteria for evaluation** Formal grade on Moodle by the supervisor. Students should present a thesis document and a scientific paper (about the student's thesis)

A i) Thesis Document (all perfect)
ii) Scientific paper (either published/accepted in A venues in the Faculty ranking or endorsed by two Faculty members of IU as A level paper).

B i) Thesis Document (with minor comments)
ii) Scientific paper (either published/accepted in B venues in the Faculty ranking or endorsed by two Faculty members of IU as B level paper).

C (Anything else acceptable)

D (Fail)

### G.11.5 Panel of experts

Demo and presentation in front of a panel of experts. Deadline sometime in XXX

# Appendix H

# Knowledge Areas

- Intelligent Systems (IS)
- Graphics & Visualization (GV)
- Programming Languages (PL)
- Parallel & Distributed Programming (PD)
- Software Development Fundamentals (SDF)
- Software Engineering (SE)
- Electrical Engineering (EE)
- Control Engineering (CE)
- Operating Systems (OS)
- Computational Science (CN)
- Mathematical Physics (MP)
- Mechanical Engineering (ME)
- Algorithms and Complexity (AL)
- Human Computer Interaction (HCI)
- Social Issues and Professional Practice (SP)
- Communications (CM)

# Appendix I

# Detailed Breakdown of Knowledge Areas of the MS Degree

Tables I.1, I.2, I.3, and I.4 contain the breakdown of the knowledge areas covered in each core course of instruction in the curriculum of Software Engineering, Robotics, Data Science, and Secure Systems and Network Engineering respectively.

**Table I.1:** Breakdown of the knowledge areas covered in each core course of the curriculum in Software Engineering

| Course | SDF | SE | Ph | AL | HCI | SP |
|---|---|---|---|---|---|---|
| Personal Software Process | | 100% | | | | |
| Requirements Engineering | | 100% | | | | |
| Models of Software Systems | | 90% | | 10% | | |
| Managing Software Development | | 100% | | | | |
| Analysis of Software Artifacts | 20% | 60% | | | 20% | |
| Architectures of Software Systems | | 90% | | | 10% | |
| Metrics and empirical methods for software engineers and data scientists | 20% | 80% | | | | |
| Communication | | | 20% | | | 80% |

**Table I.2:** Breakdown of the knowledge areas covered in each core course of the curriculum in Robotics

| Course | IS | GV | EE | CE | ME | AL |
|---|---|---|---|---|---|---|
| Sensation, perception, and actuation | 40% | | 40% | 20% | | |
| Advanced robotics | 40% | | | 20% | 40% | |
| Machine Learning | 60% | | | | | 40% |
| Dynamics of Nonlinear Robotic Systems | 30% | | | 30% | 40% | |
| Advanced Computer Vision | 60% | 20% | | | | 20% |
| Computational Intelligence | 60% | | | 20% | | 20% |
| Behavioral and cognitive robotics | 50% | | | | | 50% |
| Neuroscience | 60% | | | 20% | | 20% |

**Table I.3:** Breakdown of the knowledge areas covered in each core course of the curriculum in Data Science

| Course | IS | GV | SE | CN | AL | IM |
|---|---|---|---|---|---|---|
| Metrics and empirical methods for software engineers and data scientists | | | 80% | | 20% | |
| High-Dimensional Data Analysis | 40% | 20% | | | 40% | |
| Advanced Information Retrieval | 20% | | | 20% | 20% | 40% |
| Big Data Technologies and Analytics | 20% | | | 20% | 20% | 40% |
| Advanced Statistics | | | | 100% | | |
| Machine Learning | 80% | | | 10% | 10% | |
| Advanced Machine Learning | 80% | | | 10% | 10% | |
| Optimization | 20% | | | 20% | 60% | |
| Managing Software Development | | | 100% | | | |

**Table I.4:** Breakdown of the knowledge areas covered in each core course of the curriculum in Secure Systems and Network Engineering

| Course | PL | PD | OS | CN | AL | CM |
|---|---|---|---|---|---|---|
| Advanced Networking | | | | | 30% | 70% |
| Classical Internet Applications | | | | | 40% | 60% |
| Distributed Systems | | 10% | 20% | | 40% | 30% |
| Security of System and Networks | | | | | 50% | 50% |
| Cybercrimes and Forensics | | | | 20% | | 80% |
| Advanced Security | | | 20% | | 30% | 50% |
| Large Installation Administration | 10% | | 20% | 10% | 25% | 35% |
| Offensive Technologies | 10% | | 5% | | 25% | 60% |

# Appendix J

# Languages, Technologies, and Tools used in the Courses during the MS degree

In the context of the courses, several (programming) languages, technologies, and tools are used. Here below there is a list of some of the most relevant.

## J.1 Software Engineering

### J.1.1 (Programming) Languages

- bash scripting language
- C
- C++
- Java
- Python
- R
- SQL
- UML

### J.1.2 Tools and Technologies

- Apache Cassandra (https://cassandra.apache.org/)
- Apache Hadoop (http://hadoop.apache.org/) with MapReduce (https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html)
- Apache Spark (https://spark.apache.org/)
- Docker (https://www.docker.com/)
- Eclipse (http://www.eclipse.org/)
- JUnit (http://www.junit.org/)
- WEKA (https://www.cs.waikato.ac.nz/ml/weka/)

In addition, the students experiment the Windows and Linux operating system families, in their most common versions.

## J.2 Robotics

### J.2.1 (Programming) Languages

- bash scripting language
- C
- C++
- Java
- Python

### J.2.2 Tools and Technologies

- Eclipse (http://www.eclipse.org/)
- Matlab / Simulink (https://www.mathworks.com/products/simulink.html)
- Octave (https://www.gnu.org/software/octave/)
- OpenCV (https://opencv.org/)
- Scikit-learn (http://scikit-learn.org/)
- Scilab/Xcos (https://www.scilab.org/en)
- WEKA (https://www.cs.waikato.ac.nz/ml/weka/)

In addition, the students experiment the Windows and Linux operating system families, in their most common versions.

## J.3 Data Science

### J.3.1 (Programming) Languages

- bash scripting language
- C
- C++
- Java
- Python
- R
- SQL

### J.3.2 Tools and Technologies

- Anaconda (https://anaconda.org/)
- Apache Hadoop (http://hadoop.apache.org/) with MapReduce (https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html)
- Docker (https://www.docker.com/)
- Eclipse (http://www.eclipse.org/)
- Keras (https://keras.io/)
- NoSQL: Graph DB (https://neo4j.com/developer/)
- Numpy (http://www.numpy.org/)
- Orange (https://orange.biolab.si/)
- Pandas (https://pandas.pydata.org/)
- Pomegranade (http://www.pomegranateapps.com/mac/)
- PyCharm (https://www.jetbrains.com/pycharm/)
- Scikit-learn (http://scikit-learn.org/)
- Scilab/Xcos (https://www.scilab.org/en)
- WEKA (https://www.cs.waikato.ac.nz/ml/weka/)
- Tensorflow (https://www.tensorflow.org/)

In addition, the students experiment the Windows and Linux operating system families, in their most common versions.

## J.4 Secure Systems and Network Engineering

### J.4.1 (Programming) Languages

- bash scripting language
- C
- C++
- Java
- Python

### J.4.2 Tools and Technologies

- Apache Cassandra (`https://cassandra.apache.org/`)
- Apache Hadoop (`http://hadoop.apache.org/`) with MapReduce (`https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html`)
- Apache Spark (`https://spark.apache.org/`)
- Docker (`https://www.docker.com/`)
- Eclipse (`http://www.eclipse.org/`)
- Networkx (`https://networkx.github.io/`)
- Wireshark (`https://www.wireshark.org/`)
- *nix command-line (traceroute, ping, ssh, flex, bison ...)

In addition, the students experiment the Windows and Linux operating system families, in their most common versions.

# Appendix K

# Catalogue of the Core Courses

## K.1 Introduction to Programming I

- **Course name:** Introduction to Programming I
- **Course number:** XYZ
- **Subject area:** Programming Languages and Software Engineering

### K.1.1 Course characteristics

#### K.1.1.1 Key concepts of the class

- Basic concept - algorithm, program, data
- Computer architecture basics
- Structured programming
- Object-oriented programming
- Generic programming
- Exception handling
- Programming by contract (c)
- Functional programming
- Concurrent programming

#### K.1.1.2 What is the purpose of this course?

The Introduction to Programming course teaches the fundamental concepts and skills necessary to perform programming at a professional level. Students will learn how to master the fundamental control structures, data structures, reasoning patterns and programming language mechanisms characterizing modern programming, as well as the fundamental rules of producing high-quality software. They will acquire the necessary programming background for later courses introducing programming skills in specialized application areas. The course focuses on Object Oriented paradigm.

#### K.1.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

- Basic concepts of programming. What is algorithm, program.
- Concept of typification. Dynamic and static types.
- Concepts of structured programming, object-oriented one.
- Concepts of exception handling and generic programming.
- Concurrent programming and functional programming in imperative programming languages.
- verification of the software based on programming by contract (C)

**- What should a student be able to understand at the end of the course?**

- How to create high quality software using mainstream concepts of programming.
- What is object-oriented programming and its main advantages
- How to increase the level of abstraction with help of genericity.
- How to create concurrent programs and what are the main issues related to this kind of programming

**- What should a student be able to apply at the end of the course?**

- To be able to create quality programs in Java.

#### K.1.1.4 Course evaluation

**Table K.1:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 40 | 40 |
| Interim performance assessment | 30 | 30 |
| Exams | 30 | 30 |

If necessary, please indicate freely your course's features in terms of students' performance assessment:

**Labs/seminar classes:**

- In-class participation 1 point for each individual contribution in a class but not more than 1 point a week (i.e. 14 points in total for 14 study weeks),
- overall course contribution (to accumulate extra-class activities valuable to the course progress, e.g. a short presentation, book review, very active in-class participation, etc.) up to 6 points.

**Interim performance assessment:**

- in-class tests up to 10 points for each test (i.e. up to 40 points in total for 2 theory and 2 practice tests),
- computational practicum assignment up to 10 points for each task (i.e. up to 30 points for 3 tasks).

**Exams:**

- mid-term exam up to 30 points,
- final examination up to 30 points.

**Overall score:** 100 points (100%).

**Table K.2:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 85-100 | 85-100 |
| B. Good | 75-84 | 75-84 |
| C. Satisfactory | 60-75 | 60-75 |
| D. Poor | 0-59 | 0-59 |

#### K.1.1.5 Grades range

If necessary, please indicate freely your course's grading features:

- A: more than 85 of the overall score;
- B: at least 85 of the overall score;
- C: at least 75 of the overall score;
- D: less than 60 of the overall score.

#### K.1.1.6 Resources and reference material

**Textbook:**

- 
- 

### K.1.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.3:** Course Sections

| Section Number | Section Title | Lectures (hours) | Seminars (labs) | Self-study | Knowledge evaluation |
|---|---|---|---|---|---|
| 1 | Introduction to programming | 12 | 6 | 12 | 2 |
| 2 | Introduction to object-oriented programming | 8 | 4 | 8 | 1 |
| 3 | Introduction to generics, exception handling and programming by contract (C) | 8 | 4 | 8 | 1 |
| 4 | Introduction to programming environments | 12 | 6 | 12 | 2 |
| 5 | Introduction to concurrent and functional programming | 8 | 4 | 8 | 1 |
| Final examination | | | | | 2 |

#### K.1.2.1 Section 1

**Section title:** Introduction to programming

**Topics covered in this section:**

- Basic definitions – algorithm, program, computer, von Neumann architecture, CPU lifecycle.
- Programming languages history and overview. Imperative (procedural) and functional approaches.
- Translation – compilation vs. interpretation. JIT, AOT. Hybrid modes.
- Introduction to typification. Static and dynamic typing. Type inference. Basic types – integer, real, character, boolean, bit. Arrays and strings. Records-structures.
- Programming – basic concepts. Statements and expressions. 3 atomic statements - assignment, if-check, goto. Control structures – conditional, assignment, goto, case-switch-inspect, loops.
- Variables and constants.
- Routines – procedures and functions.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the difference between compiler and interpreter?
2. What is the difference between type and variable?
3. What is the background of structured programming?

**Typical questions for seminar classes (labs) within this section**

1. How to compile a program?
2. How to run a program?
3. How to debug a program?

**Test questions for final assessment in this section**

1. What are the basic control structure of structured programming?
2. What is the difference between statements and expressions?
3. What are the benefits of type inference?

#### K.1.2.2 Section 2

**Section title:** Introduction to object-oriented programming

**Topics covered in this section:**

- Key principles of object-oriented programming
- Overloading is not overriding
- Concepts of class and object
- How objects can be created?
- Single and multiple inheritance

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the meaning of polymorphism?
2. How to check the dynamic type of an object?
3. What are the limitations of single inheritance?
4. What are the issues related with multiple inheritance?

**Typical questions for seminar classes (labs) within this section**

1. How to handle array of objects of some class type?
2. How to implement the class which logically has to have 2 constructors with the same signature but with different semantics?

**Test questions for final assessment in this section**

1. Name all principles of object-oriented programming?
2. Explain what conformance means?
3. Explain why cycles are prohibited in the inheritance graph?

#### K.1.2.3 Section 3

**Section title:** Introduction to generics, exception handling and programming by contract (C)

**Topics covered in this section:**

- Introduction to generics

- Introduction to exception handling
- Introduction to programming by contract (C)

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is constrained genericity?
2. What is exception?
3. What is assertion?

**Typical questions for seminar classes (labs) within this section**

1. How constrained genericity may be used for sorting of objects?
2. In which order catch blocks are being processed?
3. Where is the problem when precondition is violated?

**Test questions for final assessment in this section**

1. Can array be treated as generic class?
2. What is the difference between throw and throws in Java?
3. What is purpose of the class invariant?

#### K.1.2.4 Section 4

**Section title:** Introduction to programming environments

**Topics covered in this section:**

- Concept of libraries as the basis for reuse.
- Concept of interfaces/API. Separate compilation.
- Approaches to software documentation.
- Persistence. Files.
- How to building a program. Recompilation problem. Name clashes, name spaces

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How reuse helps to develop software?
2. How concept of libraries and separate compilation co-relate?
3. What are the benefits of integrating documentation into the source code?
4. Why is it essential to have persistent data structures?

**Typical questions for seminar classes (labs) within this section**

1. What is to be done to design and develop a library?
2. How to add documenting comments into the source code?
3. What ways exists in Java to support persistence ?

**Test questions for final assessment in this section**

1. How to deal with name clashes?
2. What is the main task of the recompilation module?
3. What are the differences between different formats of persistence files?

#### K.1.2.5 Section 5

**Section title:** Introduction to concurrent and functional programming

**Topics covered in this section:**

- Concurrent programming.
- Functional programming within imperative programming languages.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Explain the key differences parallelism and concurrency
2. What are the key issues related to parallel execution?
3. What are the models of parallel execution?
4. What is the difference between function and object?

**Typical questions for seminar classes (labs) within this section**

1. Which Java construction support concurrency?
2. What is a thread?
3. What is in-line lambda function?

**Test questions for final assessment in this section**

1. What is the meaning of SIMD and MIMD?
2. What are the implications of the Amdahl's law?
3. What model of concurrency Java relies on?

4. Which function can be considered as pure?
5. How to declare a function to accept a functional object as its argument?
6. How Java supports high-order functions?
7. How capturing variables works in Java?

## K.2 Computer Architecture

- **Course name:** Computer Architecture
- **Course number:** XXX

### K.2.1 Course Characteristics

#### K.2.1.1 Key concepts of the class

- The fundamental principles for modern computer systems
- Computer instructions, their representation, and execution
- Computer arithmetics

#### K.2.1.2 What is the purpose of this course?

The course covers the fundamental principles of computer systems design. We first overview the key hardware components of a modern computer system, available performance metrics, and the general principles of computer architecture. We then discuss the representation and execution of computer instructions, instruction set architecture, the translation hierarchy of a high-level program into machine code. We also cover the elements of computer arithmetics, logic circuits, including combinational and sequential logic circuits. These theoretical principles are illustrated by using MIPS instruction set architecture, FPGA, and Verilog HDL programming language during the labs. We then study in details simple and pipelined implementation schemes of a processor, the idea of a pipelined execution, related hazards and their solutions. We complete the course by introducing several advanced topics, including computer security and vulnerabilities, GPU programming, and modern principles for memory hierarchy design.

#### K.2.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

- Key components of a modern computer system;
- Available performance metrics for computer systems;

- Computer arithmetics operations, including floating point numbers;
- Number systems and conversion between them;
- Representation formats for computer instructions

**- What should a student be able to understand at the end of the course?**

- Fundamental principles of computer architecture (Moore's law, memory hierarchy, multiprocessing, speculative execution, and others);
- The design scheme of a modern processor;
- The interaction principles between software and hardware;
- Program representation and execution by a computer system

**- What should a student be able to apply at the end of the course?**

- The design skills of logic circuits by using Verilog HDL programming language;
- FPGA programming by using Quartus Prime software;
- MIPS assembly programming (including MARS simulator);

#### K.2.1.4 Course evaluation

**Table K.4:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 20 |
| Interim performance assessment | 30 | 40 |
| Exams | 50 | 40 |

#### K.2.1.5 Grades range

#### K.2.1.6 Resources and reference material

- 
- 
- Handouts supplied by the instructor
- Online resources and research papers shared by instructor

**Table K.5:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 90-100 |
| B. Good | 75-89 | 70-89 |
| C. Satisfactory | 60-74 | 60-69 |
| D. Poor | 0-59 | 0-59 |

## K.2.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.6:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction to the Fundamental Concepts of Computer Architecture | 18 |
| 2 | Computational Logic Implementation in a Computer System | 15 |
| 3 | Instruction Representation and Execution in a Computer System | 15 |
| 4 | Computer Arithmetics | 15 |
| 5 | Processor Architecture | 15 |
| 6 | Advanced Topics | 6 |

### K.2.2.1 Section 1

**Section title:** Introduction to the Fundamental Concepts of Computer Architecture

**Topics covered in this section:**

- Key Components of a Computer System
- Fundamental Ideas of Computer Architecture
- Translation Hierarchy of a High-Level Program into Machine Code
- Performance Metrics of a Computer System

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**
Sample quistions from weekly quizzes:

1. Do you agree that main memory (RAM) is a non-volatile memory?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

2. There are several types of memory available for computers, such as CPU cache, main memory (RAM), SSD, etc. What are the key differences between them?
3. What is the key principle behind the Von Neumann Architecture?
4. Specify a correct order for tools used during high-level program translation and execution: Compiler, Assebler, Linker, Loader;
5. Let a program run on a computer comprised of one processor only. Let us now increase the number of processors up to m>1, so that multiple instructions of that program can be executed in parallel. Assume that all processor speeds are the same. Do you agree that a program can never execute slower on m processors, as compared to the case of one processor?

**Typical questions for seminar classes (labs) within this section**

1. Demonstration and description of key elements of an FPGA board (memory unit, PCI slot, clock generator, etc.);
2. Description of specific features of FPGA as compared to other integrated circuit devices;
3. Writing basic code for FPGA board;
4. Configuration and usage of the basic functionality in Quartus Prime software

**Test questions for final assessment in this section**

1. Briefly describe the principles of Von Neumann architecture. Illustrate with a diagram.
2. Describe the steps that transform a program written in a high-level language such as C into a representation that is directly executed by a computer processor. Illustrate with a diagram; provide a brief description for each step.
3. Consider three different processors P1, P2, and P3 executing the same instruc-

tion set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0. P3 has a 4.0 GHz clock rate and has a CPI of 2.2. Answer the following questions: a) Which processor has the highest performance expressed in instructions per second? b) If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions. c) We are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

#### K.2.2.2 Section 2

**Section title:** Computational Logic Implementation in a Computer System

**Topics covered in this section:**

- Logic Gates and Boolean Algebra
- Logic Circuits
- Combinational and Sequential Logic
- Number Systems
- The Basics of Verilog Hardware Description Language (HDL) Programming

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

Sample questions from weekly quizzes:

1. Convert decimal number 123 into base-5 format;
2. Do you agree that a S/R latch and a D flip-flop have different storage capacities?
3. Choose the key differences between SRAM and DRAM memory types: cost, power consumption, volatility, access speed, storage capacity, etc.;
4. Do you agree that one of the key differences between sequential and combinational logic circuits is the presence of memory elements?

**Typical questions for seminar classes (labs) within this section**

1. Questions regarding the basic logic gates;
2. Assignments to design simple logic circuits with 2-3 logic gates on a white board;
3. Programming assignments in Quartus Prime software, to design and compile simple logic circuits;
4. Programming an FPGA board by using Verilog HDL in Quartus Prime environment, such as turning on or off leds based on a switch position;
5. Questions regarding the difference between combinational and sequential logic circuits;

**Test questions for final assessment in this section**

1. Prove that the AND and NOT logic gates can be implemented by using only the NOR logic gate.
2. What are the S/R latch and D latch? Draw the respective logic circuits. Describe the differences between them.
3. Briefly describe the key difference(s) between combinational and sequential logic circuits.
4. Define what a multiplexor logic circuit is (with an arbitrary number of inputs). Provide a truth table for a 2-to-1 multiplexor. Provide a logic circuit implementing a 2-to-1 multiplexor, that uses AND, NOT, and OR logic gates. Describe a Verilog module implementing such a logic circuit of a 2-to-1 multiplexor.

#### K.2.2.3 Section 3

**Section title:** Instruction Representation and Execution in a Computer System

**Topics covered in this section:**

- Instruction Set Architecture (ISA)
- The Overview of MIPS ISA
- Types of MIPS Instructions and Their Representation in a Binary Format
- Sample MIPS Assembly Programs

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**
Sample questions from weekly quizzes:

1. How many bits are in one MIPS word?
2. Which MIPS directive would you use to create a string data?
3. For MIPS instruction set architecture (ISA), each register is reserved for a specific purpose. Describe the purpose of registers listed below: $v0, $s0-$s7, $t0-$t7;
4. In MARS simulator for MIPS programming, all register values, that are displayed in the register viewer, start with prefix "0x". What is the meaning of this prefix?

**Typical questions for seminar classes (labs) within this section** Sample MIPS programming assignments in MARS simulator:

1. Print a "Hello, World!" message in a console;
2. Computation of a simple arithmetic expression for integer parameters;
3. Computation of the first 10 Fibonacci numbers;
4. Implementation of more advanced program structures, such as conditional loops

**Test questions for final assessment in this section**

1. Translate the following MIPS code to C (or pseudocode). Assume that variables f, g, h, and i are assigned to registers $s0, $s1, $s2, and $s3, respectively. Code to translate: sub $t0, $s1, $s2; addi $t0, $t0, 3; add $s0, $s3, $t0
2. Assume that two MIPS registers, $s0ands1$, contain the following binary data (for simplicity, we assume 8-bit registers, rather that 32): $s0: 00100000; $s1: 01010101. What is the value of register $s1 after the execution of the following MIPS instruction?: sll $s1, $s0, 4.
3. List and describe the purpose of general-purpose MIPS registers.

#### K.2.2.4 Section 4

**Section title:** Computer Arithmetics

**Topics covered in this section:**

- Basic Arithmetic Operations (Bitwise, Shifts, Multiplication, Division, and Others)
- Overflow and Underflow Problems for Arithmetic Operations
- Arithmetic Operations with Floation Point Numbers
- Problems Related to Precision and Conversion for Floating Point Numbers

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

Sample questions from weekly quizzes:

1. Assume that two MIPS registers, \$s0 and \$s1, contain the following binary data: \$s0: 00100000; \$s1: 01010101 (For simplicity, we assume 8-bit registers, rather that 32) What is the value of \$s1 after the execution of the following instruction?: sll \$s1, \$s0, 4
2. What is a "register spilling" in the context of MIPS instruction set architecture?
3. Do you agree with the following statement? In some cases, MIPS logical shift operations, sll and srl, can be used as an efficient alternative to multiplication and division operations, mul and div.
4. Do you agree that overflow and underflow exceptions correspond to cases, when the result of an arithmetic operation surpasses and subceeds, respectively, the maximum and the minimum value for an appropriate data type returned by that arithmetic operation?

**Typical questions for seminar classes (labs) within this section** Sample MIPS programming assignments in MARS simulator, to practice floating-point operations:

1. Division of two floating-point numbers;
2. Conversion of Fahrenheit into Celsius temperature, and vice versa;
3. Computation of a sphere surphase area;
4. Questions regarding the execution of arithmetic operations with interger and floating-point values

**Test questions for final assessment in this section**

1. Briefly describe the overflow and underflow problems for arithmetic operations.
2. Describe the difference between executing arithmetic operations with integers and floating-point values for a MIPS processor.
3. What is a precision problem for a floating-point operation?

#### K.2.2.5 Section 5

**Section title:** Processor Architecture

**Topics covered in this section:**

- Key Components of a Processor: Control and Arithmetic Logic Unit, Registers
- Processor Datapath and Control Signals
- The Notion of a Pipelined Execution, Pipeline Hazards, and Their Solutions
- A Simple and Pipelined Implementation Schemes of a Processor

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

Sample questions from weekly quizzes:

1. Do you agree that the key motivation for the CPU pipelining is to speed-up the execution of a program by exploring multiple CPU cores?
2. Which CPU block(s) is/are accessed during the execution of the following instruction? lw $1, 5($2)
3. What are 5 major stages of a pipelined instruction execution?
4. Do you agree that, for a processor with 5 pipelined stages, the number of concurrently executed instructions is up to 4?
5. There are several types of processors available, including single-cycle and multicycle.The major advantage of a single-cycle processor is the simplicity of its design. But what is its key drawback?

**Typical questions for seminar classes (labs) within this section**

1. Design of a testbench in ModelSim for Quartus Prime programming environment;
2. The design of Half-Adder, Full-Adder, Ripple Carry Adder by using Verilog HDL in Quartus Prime
3. Testing the correctness of Verilog HDL design by using ModelSim

**Test questions for final assessment in this section**

1. What is a Program Counter (PC) register of a processor?
2. Describe the principle of a pipelined CPU execution. Provide a diagram illustrating the concept. Briefly describe the 5 key stages of a classical pipeline.
3. What are the key differences between Control Unit (CU) and Arithmetic Logic Unit (ALU) of a processor? Which purposes do they serve?
4. What is a CPU datapath?

#### K.2.2.6 Section 6

**Section title:** Advanced Topics

**Topics covered in this section:**

- Computer Security and Vulnerabilities
- Graphics Processing Unit (GPU) and General-Purpose GPU Programming
- Modern Approaches for Memory Hierarchy Design

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Cold boot attack explores vulnerabilities in a memory dump mechanism. What is a memory dump?
2. Below is a list of possible vulnerability attacks. Choose the one(s) that explore(s) vulnerabilities in a speculative execution of modern processors: Meltdown, Foreshadow, Cold boot attack, Spectre, No choice is correct;
3. Choose the most precise definition for a side-channel attack: An attack that explores vulnerabilities in the hardware implementation of a computer system, An attack that explores vulnerabilities in the software components of a computer system;
4. Do you agree that Meltdown and Spectre vulnerabilities both explore race conditions in existing memory circuits?

**Typical questions for seminar classes (labs) within this section**

1. Programming assignment to implement Multiplexor using Verilog HDL in Quartus Prime;
2. Performance optimization of a Verilog HDL design;
3. The design of a simple Arithmetic-Logic Unit (ALU);
4. Revision questions

**Test questions for final assessment in this section**

1. What is an out-of-order execution? What hardware features of CPU implementation, in addition to an out-of-order execution, are exploited by Meltdown vulnerability? How serious is Meltdown vulnerability?
2. What is an instruction-level parallelism?
3. Describe the idea of a general-purpose GPU programming.
4. Briefly explain the working principles of a CPU cache.
5. Discuss advantages and drawbacks of a hierarchical memory model for computer systems.

## K.3 Discrete Mathematics

- **Course name:** Discrete Mathematics
- **Course number:** XYZ
- **Subject area:** Mathematics

### K.3.1 Course characteristics

#### K.3.1.1 Key concepts of the class

- Basic proof techniques (by construction, by induction, by contradiction)
- sets and naíve set theory, relations and functions
- finite combinatorics and discrete probability, discrete optimization
- directed and undirected graphs, graph traverses and trees
- lattices and Boolean algebras;
- propositional logic, propositional modelling, propositional satisfiability problem

#### K.3.1.2 What is the purpose of this course?

The course is designed for Software Engineering and Computer Science students to teach them basic reasoning and proving skills and provide them by a correct knowledge of basic (core) concepts, definitions, theoretical results and elementary applied methods and techniques of the set theory, algebra of relations and functions, finite combinatorics and discrete probability, graph theory, discrete optimization and dynamic programming, lattice theory and Boolean algebras, Boolean logic and propositional calculus, propositional modelling and satisfiability. All definitions and theorem statements (that will be given in lectures and that are needed to explain the keywords listed above) will be formally stated, but just few of them will be formally proven. Instead (in practice classes) we will try these definitions and theorems on work with routine exercises and a little bit more complicated (complex) problems.

#### K.3.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

- elements and basics of the naïve set theory,
- basic concepts of relations and functions,
- basic formulas of enumerating Combinatorics,
- basic definitions and properties of graph theory,
- propositional logic and reasoning,
- match the concrete numerical approach with the necessary level of accuracy.

**- What should a student be able to understand at the end of the course?**

- role of proof techniques for sound reasoning in mathematics
- role of set theory in foundations of mathematics
- role of discrete mathematics for foundations of programming
- role of discrete mathematics for modeling of discrete systems

**- What should a student be able to apply at the end of the course?**

- basic graph algorithms (path-finding, drawing, traversing)
- basic algorithm design techniques (backtracking, branch-and-bound, dynamic programming)
- basic propositional reasoning algorithms (satisfiability, tautology, derivation)

#### K.3.1.4 Course evaluation

**Table K.7:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 20 |
| Interim performance assessment | 30 | 40 |
| Exams | 50 | 40 |

If necessary, please indicate freely your course's features in terms of students' performance assessment:

**Labs/seminar classes:**

- In-class participation 1 point for each individual contribution in a class but not more than 1 point a week (i.e. 14 points in total for 14 study weeks),
- overall course contribution (to accumulate extra-class activities valuable to the course progress, e.g. a short presentation, book review, very active in-class participation, etc.) up to 6 points.

**Interim performance assessment:**

- Each of 4 (2 theory and 2 practice) in-class tests costs 10 points.

**Exams:**

- mid-term exam and final examination costs up to 20 points each (i.e. 40 points for both).

**Overall score:** 100 points (100%).

#### K.3.1.5 Grades range

**Table K.8:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 70-79 |
| C. Satisfactory | 60-74 | 60-69 |
| D. Poor | 0-59 | 0-59 |

If necessary, please indicate freely your course's grading features:

- A: at least 80% of the overall score;
- B: at least 70% of the overall score;
- C: at least 60% of the overall score;
- D: less than 60% of the overall score.

#### K.3.1.6 Resources and reference material

**Textbook:**

•

**Reference material:**

•
•

### K.3.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.9:** Course Sections

| Section Number | Section Title | Lectures (hours) | Seminars (labs) | Self-study | Knowledge evaluation |
|---|---|---|---|---|---|
| 1 | Basic proof techniques and the naïve set theory | 4 | 4 | 4 | 1 |
| 2 | Basics of relations, functions and enumerating combinatorics | 6 | 6 | 6 | 2 |
| 3 | Basics of graphs theory, trees and discrete optimization | 6 | 6 | 6 | 0 |
| 4 | Boolean algebras and logic, propositional reasoning | 6 | 6 | 6 | 1 |
| Final examination | | | | | 2 |

#### K.3.2.1 Section 1

**Section title:** Basic proof techniques and the naïve set theory

**Topics covered in this section:**

- Basic proof techniques (by construction, by contradiction, by induction)
- Postulates of the naïve set theory.
- Introduction of (natural and rational) numbers in the naïve set theory.
- Principle of mathematical induction.

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**What forms of evaluation were used to test students' performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

1. Define set intersection using the postulates of the naïve set theory.
2. Compute the number of parenthesis in the numeral for natural number $n$.
3. Prove existence of the natural numbers using the naïve set theory.
4. Define addition for natural numbers using the naïve set theory.

**Typical questions for seminar classes (labs) within this section**

1. Define the standard binary set union using the naïve set theory.
2. Compute the number of subsets in the numeral for natural number $n$.
3. Define "less than" binary relation on the natural numbers using the naïve set theory.
4. Proof (using the naïve set theory) uniqueness of the natural numbers.

**Test questions for final assessment in this section**

1. Define the standard binary Cartesian product using the naïve set theory.
2. Compute the length of the representation of the numeral for natural number $n$.
3. Define multiplication on the natural numbers using the naïve set theory.
4. Proof existence of rational numbers using the naïve set theory.

### K.3.2.2 Section 2

**Section title:** Basics of relations, functions and enumerating combinatorics

**Topics covered in this section:**

- Relations and functions, operations and relations over a set

- Algebra of binary relations
- Denotational semantics of a simple imperative programming language
- Three main principles of finite combinatorics
- Arrangements, permutations, combinations and some other selected combinatoric formulas

### What forms of evaluation were used to test students' performance in this section?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

### Typical questions for ongoing performance evaluation within this section

1. Compute the number of partial functions from a finite set with $n$ elements to a finite set with $m$ elements.
2. What is the decimal size of $2_{10}^{100}$?
3. Show that the number of compositions of $n \geq 0$ with exactly $k \geq 0$ addends is $C_{n-1}^{k-1}$.
4. Course has 90 enrolled students, a project group must consist of 3 students, groups will be formatted in random. Explain what does this statement mean? (I.e. explain what are $\Omega$, $\mathscr{F}$, and $P$ in the definition of the probability space.)

### Typical questions for seminar classes (labs) within this section

1. Let $n, k > 0$ be integer; explain (don't prove!) using intuitive set theory why
   - $C_n^k = C_{n-1}^{k-1} + C_{n-1}^k$;
   - $\sum_{m=0}^{m=n} C_n^m \ = 2^n$.
2. Evaluate (without use of a computer) the decimal size of 100! (i.e. provide reliable/provable lower and upper bound for the length of the decimal representation of this number).
3. Prove that in the denotational semantics programs $\big(\alpha; (\beta; \gamma)\big)$ and $\big((\alpha; \beta); \gamma\big)$.
4. Prove that the reflexive-transitive closure $R^*$ of a binary relation $R$ is the least reflexive and transitive binary relation that contains $R$.

### Test questions for final assessment in this section

1. Characterize in the terms of arity, reflexivity, symmetry, etc.,

- relations “a number is prime”, “numbers are co-prime”, “a number is a common multiple of two numbers” on natural numbers;
- relations “less than”, “not-greater than”, etc., on natural numbers;
- relations “less than”, “not-greater than”, etc., on rational and real numbers.

2. Prove the following equalities in any algebra of binary relations:
   - $(P \circ Q) \circ R = P \circ (Q \circ R)$;
   - $(P \circ Q)^{-} = Q^{-} \circ P^{-}$;
   - $(P \cup Q)^{-} = P^{-} \cup Q^{-}$.
3. Define (separately) antisymmetric, symmetric-, reflexive-, transitive- closures of a binary relation.
4. Explain (using the standard set-theoretic notation and operations) how to build the above closures from the equality, $R$ and the inverse $R^{-}$.

#### K.3.2.3 Section 3

**Section title:** Basics of graphs theory, trees and discrete optimization

**Topics covered in this section:**

- Definition and main properties of graphs and digraphs.
- Selected classes of (di)graphs (empty, complete, chains, etc).
- Euler graphs.
- Graph complements and “almost all graphs”.
- Plane and planar graphs, Euler formula.
- Graph traversing, (directed) trees, König Lemma.

**What forms of evaluation were used to test students’ performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Draw complete graphs of orders 2, 3, 4 and 6.
2. How many Petri nets (without edge multiplicities) with $m$ places and $n$ transitions do exist?
3. Check Euler's formula for the regular dodecahedron.
4. A terminal node of a graph is a node with degree 1. Prove that a tree has at least 2 terminal nodes.

**Typical questions for seminar classes (labs) within this section**

1. How many edges has the complete graph of order $n$? How many edges has the complete digraph of order $n$?
2. Expend Euler theorem (about Euler paths) on multi-graphs.
3. Prove that $K_5$ and $K_{3,3}$ aren't planar graphs.
4. A spanning tree of a graph is a tree that contains all nodes of the graph.
   - Prove that any spanning tree of a connected $(mn$-graph results after removal of $(m-n+1)$ edges.
   - Does removal of any set consisting of $(m-n+1)$ edges results in a tree?

**Test questions for final assessment in this section**

1. How many edges has complete biograph $K_{m,n}$?
2. Check that the graphs $K_1$, $P_4$, and $C_5$ are self-complementary.
3. Prove that every finite planar graph has the average node degree is less than 6.
4. Draw all (a) forests and (b) trees of orders from 1 to 6.

#### K.3.2.4 Section 4

**Section title:** Boolean algebras and logic, propositional reasoning

**Topics covered in this section:**

- Pre-order, order, linear order; lattices, complete lattices and Boolean algebras.
- Graph representation of ordered sets; topological sorting.
- Introduction to Formal Concept Analysis.

- Semantic (Boolean algebra) and syntactic (rewriting and axiomatic) approaches to tautologies.
- Boolean satisfiability, DPLL-algorithm and Boolean modelling.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Every set $S$ with a partial order $R$ is a D.A.G. $(S,R)$; every D.A.G. $(V,E)$ can be considered as P.O.S. $(V,E^*)$.
2. Prove that the following three clauses equivalent for all lattices: (a) $a \le b$, (b) $a \wedge b = a$, and (c) $a \vee b = b$.
3. What is size of a propositional formula that expresses that a graph is $n$-colourable? (Evaluate the size in terms of number of colors, nodes and edges in the graph.) How to use this formula to compute the chromatic number of the graph?
4. Assuming finite Stone representation theorem, prove that the number of elements in every finite Boolean algebra is a power of two.

**Typical questions for seminar classes (labs) within this section**

1. Give example of a p.o.s. with minimal (maximal) elements but without the least (largest) element.
2. Exercises topological sort for a graph $Div_{[1..21]} = ([1..21], \{(m,n) : 1 \le m \le n \le 21,\ m \text{ divides } n\})$.
3. Which Boolean algebra identities are valid in every lattice? In every bounded lattice? (Either prove or refute by a counterexample.)
4. Explain how DPLL algorithm may be used to check whether a propositional formula is a tautology or isn't.

**Test questions for final assessment in this section**

1. Prove that in every lattice every finite non-empty set of elements has its meet and join.
2. What is unit-clause propagation? (Give example.)
3. Explain why DPLL method is an instance of backtracking algorithm.
4. Encode 2-colourability of $K_{(3,3)}$ in Boolean satisfiability and check this formula satisfiability using DPLL.

# K.4 Mathematical Analysis I

- **Course name:** Course Title
- **Course number:** BS-MA1

## K.4.1 Course Characteristics

### K.4.1.1 Key concepts of the class

- Differentiation
- Integration
- Series

### K.4.1.2 What is the purpose of this course?

This calculus course covers differentiation and integration of functions of one variable, with applications. The basic objective of Calculus is to relate small-scale (differential) quantities to large-scale (integrated) quantities. This is accomplished by means of the Fundamental Theorem of Calculus. Should be understanding of the integral as a cumulative sum, of the derivative as a rate of change, and of the inverse relationship between integration and differentiation.

This calculus course will provide an opportunity for participants to:

- understand key principles involved in differentiation and integration of functions
- solve problems that connect small-scale (differential) quantities to large-scale (integrated) quantities
- become familiar with the fundamental theorems of Calculus
- get hands-on experience with the integral and derivative applications and of the inverse relationship between integration and differentiation.

### K.4.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to ...

- Derivative. Differential. Applications
- Indefinite integral. Definite integral. Applications
- Sequences. Series. Convergence. Power Series

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to ...

- Derivative. Differential. Applications
- Indefinite integral. Definite integral. Applications
- Sequences. Series. Convergence. Power Series
- Taylor Series

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

- Take derivatives of various type functions and of various orders
- Integrate
- Apply definite integral
- Expand functions into Taylor series
- Apply convergence tests

#### K.4.1.4 Course evaluation

**Table K.10:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | |
| Interim performance assessment | 30 | |
| Exams | 50 | |

If necessary, please indicate freely your course's features in terms of students' performance assessment.

#### K.4.1.5 Grades range

**Table K.11:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

If necessary, please indicate freely your course's grading features.

#### K.4.1.6 Resources and reference material

- Zorich, V. A. "Mathematical Analysis I, Translator: Cooke R." (2004)
-
-

### K.4.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.12:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Sequences and Limits | 28 |
| 2 | Differentiation | 24 |
| 3 | Integration and Series | 28 |

#### K.4.2.1 Section 1

**Section title:** Sequences and Limits

**Topics covered in this section:**

- Sequences. Limits of sequences
- Limits of sequences. Limits of functions
- Limits of functions. Continuity. Hyperbolic functions

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. A sequence, limiting value
2. Limit of a sequence, convergent and divergent sequences
3. Increasing and decreasing sequences, monotonic sequences
4. Bounded sequences. Properties of limits
5. Theorem about bounded and monotonic sequences.
6. Cauchy sequence. The Cauchy Theorem (criterion).
7. Limit of a function. Properties of limits.
8. The first remarkable limit.
9. The Cauchy criterion for the existence of a limit of a function.
10. Second remarkable limit.

**Typical questions for seminar classes (labs) within this section**

1. Find a limit of a sequence
2. Find a limit of a function

**Test questions for final assessment in this section**

1. Find limits of the following sequences or prove that they do not exist:
2. $a_n = n - \sqrt{n^2 - 70n + 1400}$;
3. $d_n = \left(\frac{2n-4}{2n+1}\right)^n$;
4. $x_n = \frac{(2n^2+1)^6(n-1)^2}{(n^7+1000n^6-3)^2}$.

#### K.4.2.2 Section 2

**Section title:** Differentiation

**Topics covered in this section:**

- Derivatives. Differentials
- Mean-Value Theorems
- l'Hopital's rule
- Taylor Formula with Lagrange and Peano remainders
- Taylor formula and limits
- Increasing / decreasing functions. Concave / convex functions

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. A plane curve is given by $x(t) = -\frac{t^2+4t+8}{t+2}$, $y(t) = \frac{t^2+9t+22}{t+6}$. Find
   (a) the asymptotes of this curve;
   (b) the derivative $y'_x$.
2. Derive the Maclaurin expansion for $f(x) = \sqrt[3]{1+e^{-2x}}$ up to $o\left(x^3\right)$.

**Typical questions for seminar classes (labs) within this section**

1. Differentiation techniques: inverse, implicit, parametric etc.
2. Find a derivative of a function
3. Apply Leibniz formula
4. Draw graphs of functions
5. Find asymptotes of a parametric function

**Test questions for final assessment in this section**

1. Find a derivative of a (implicit/inverse) function
2. Apply Leibniz formula Find $y^{(n)}(x)$ if $y(x) = \left(x^2-2\right)\cos 2x \sin 3x$.

3. Draw graphs of functions
4. Find asymptotes
5. Apply l'Hopital's rule
6. Find the derivatives of the following functions:
   (a) $f(x) = \log_{|\sin x|} \sqrt[6]{x^2+6}$;
   (b) $y(x)$ that is given implicitly by $x^3 + 5xy + y^3 = 0$.

#### K.4.2.3 Section 3

**Section title:** Integration and Series

**Topics covered in this section:**

- Antiderivative. Indefinite integral
- Definite integral
- The Fundamental Theorem of Calculus
- Improper Integrals
- Convergence tests. Dirichlet's test
- Series. Convergence tests
- Absolute / Conditional convergence
- Power Series. Radius of convergence
- Functional series. Uniform convergence

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Find the indefinite integral $\int x \ln\left(x + \sqrt{x^2-1}\right) dx$.
2. Find the length of a curve given by $y = \ln \sin x$, $\frac{\pi}{4} \leqslant x \leqslant \frac{\pi}{2}$.
3. Find all values of parameter $\alpha$ such that series $\sum_{k=1}^{+\infty} \left(\frac{3k+2}{2k+1}\right)^k \alpha^k$ converges.

### Typical questions for seminar classes (labs) within this section

1. Integration techniques
2. Integration by parts
3. Calculation of areas, lengths, volumes
4. Application of convergence tests
5. Calculation of Radius of convergence

### Test questions for final assessment in this section

1. Find the following integrals:
2. $\int \frac{\sqrt{4+x^2}+2\sqrt{4-x^2}}{\sqrt{16-x^4}}\,dx$;
3. $\int 2^{2x} e^x\,dx$;
4. $\int \frac{dx}{3x^2-x^4}$.
5. Use comparison test to determine if the following series converge. $\sum\limits_{k=1}^{\infty} \frac{3+(-1)^k}{k^2}$;
6. Use Cauchy criterion to prove that the series $\sum\limits_{k=1}^{\infty} \frac{k+1}{k^2+3}$ is divergent.
7. Find the sums of the following series:
8. $\sum\limits_{k=1}^{\infty} \frac{1}{16k^2-8k-3}$;
9. $\sum\limits_{k=1}^{\infty} \frac{k-\sqrt{k^2-1}}{\sqrt{k^2+k}}$.

## K.5 Analytical Geometry & Linear Algebra – I

- **Course name:** Analytical Geometry & Linear Algebra – I
- **Course number:** XYZ
- **Subject area:** Math

### K.5.1 Course characteristics

#### K.5.1.1 Key concepts of the class

- fundamental principles of vector algebra,
- concepts of basic geometry objects and their transformations in the plane and in the space

#### K.5.1.2 What is the purpose of this course?

This is an introductory course in analytical geometry and linear algebra. After having studied the course, students get to know fundamental principles of vector algebra and its applications in solving various geometry problems, different types of equations of lines and planes, conics and quadric surfaces, transformations in the plane and in the space. An introduction on matrices and determinants as a fundamental knowledge of linear algebra is also provided.

#### K.5.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

- List basic notions of vector algebra,
- recite the base form of the equations of transformations in planes and spaces,
- recall equations of lines and planes,
- identify the type of conic section,
- recognize the kind of quadric surfaces.

**- What should a student be able to understand at the end of the course?**

- explain the geometrical interpretation of the basic operations of vector algebra,

- restate equations of lines and planes in different forms,
- interpret the geometrical meaning of the conic sections in the mathematical expression,
- give the examples of the surfaces of revolution,
- understand the value of geometry in various fields of science and techniques.

**- What should a student be able to apply at the end of the course?**

- Perform the basic operations of vector algebra,
- use different types of equations of lines and planes to solve the plane and space problems,
- represent the conic section in canonical form,
- compose the equation of quadric surface.

#### K.5.1.4 Course evaluation

**Table K.13:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 10 |
| Interim performance assessment | 30 | 20 |
| Exams | 50 | 70 |

#### K.5.1.5 Grades range

**Table K.14:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 60-79 |
| C. Satisfactory | 60-74 | 40-59 |
| D. Poor | 0-59 | 0-39 |

#### K.5.1.6 Resources and reference material

**Textbooks:**

- 

**Reference material:**

- 
- 

### K.5.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.15:** Course Sections

| Section Number | Section Title | Lectures (hours) | Seminars (labs) | Self-study | Knowledge evaluation |
|---|---|---|---|---|---|
| 1 | Vector algebra | 8 | 4 | 10 | 2 |
| 2 | Introduction to matrices and determinants | 8 | 4 | 10 | 1 |
| 3 | Lines in the plane and in the space | 8 | 4 | 10 | 2 |
| 4 | Planes in the space | 8 | 4 | 10 | 1 |
| 5 | Quadratic curves | 8 | 4 | 10 | 2 |
| 6 | Quadric surfaces | 8 | 4 | 10 | 2 |
| Final examination | | | | | 2 |

#### K.5.2.1 Section 1

**Section title:** Vector algebra

**Topics covered in this section:**

- Vector spaces
- Basic operations on vectors (summation, multiplication by scalar, dot product)
- Linear dependency and in-dependency of the vectors
- Basis in vector spaces

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**What forms of evaluation were used to test students' performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

1. How to perform the shift of the vector?
2. What is the geometrical interpretation of the dot product?
3. How to determine whether the vectors are linearly dependent?
4. What is a vector basis?

**Typical questions for seminar classes (labs) within this section**

1. Evaluate $|\mathbf{a}|^2 - 2\sqrt{3}\mathbf{a}\cdot\mathbf{b} - 7|\mathbf{b}|^2$ given that $|\mathbf{a}| = 4$, $|\mathbf{b}| = 1$, $\angle(\mathbf{a}, \mathbf{b}) = 150°$.
2. Prove that vectors $\mathbf{b}(\mathbf{a}\cdot\mathbf{c}) - \mathbf{c}(\mathbf{a}\cdot\mathbf{b})$ and $\mathbf{a}$ are perpendicular to each other.
3. Bases $AD$ and $BC$ of trapezoid $ABCD$ are in the ratio of $4:1$. The diagonals of the trapezoid intersect at point $M$ and the extensions of sides $AB$ and $CD$ intersect at point $P$. Let us consider the basis with $A$ as the origin, $\overrightarrow{AD}$ and $\overrightarrow{AB}$ as basis vectors. Find the coordinates of points $M$ and $P$ in this basis.
4. A line segment joining a vertex of a tetrahedron with the centroid of the opposite face (the centroid of a triangle is an intersection point of all its medians) is called a median of this tetrahedron. Using vector algebra prove that all the four medians of any tetrahedron concur in a point that divides these medians in the ratio of $3:1$, the longer segments being on the side of the vertex of the tetrahedron.

**Test questions for final assessment in this section**

1. Vector spaces. General concepts.
2. Dot product as an operation on vectors.
3. Basis in vector spaces. Its properties.

#### K.5.2.2 Section 2

**Section title:** Introduction to matrices and determinants

**Topics covered in this section:**

- Relationship between Linear Algebra and Analytical Geometry
- Matrices 2x2, 3x3
- Determinants 2x2, 3x3
- Operations om matrices and determinants
- The rank of a matrix
- Inverse matrix
- Systems of linear equations
- Changing basis and coordinates

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the difference between matrices and determinants?
2. Matrices $A$ and $C$ have dimensions of $m \times n$ and $p \times q$ respectively, and it is known that the product $ABC$ exists. What are possible dimensions of $B$ and $ABC$?
3. How to determine the rank of a matrix?
4. What is the meaning of the inverse matrix?
5. How to restate a system of linear equations in the matrix form?

**Typical questions for seminar classes (labs) within this section**

Let $A = \begin{pmatrix} 3 & 1 \\ 5 & -2 \end{pmatrix}$, $B = \begin{pmatrix} -2 & 1 \\ 3 & 4 \end{pmatrix}$, and $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

1. Find $A + B$ and $2A - 3B + I$.
2. Find the products $AB$ and $BA$ (and so make sure that, in general, $AB \neq BA$ for matrices).
3. Find the inverse matrices for the given ones.
4. Find the determinants of the given matrices.
5. Point $M$ is the centroid of face $BCD$ of tetrahedron $ABCD$. The old coordinate system is given by $A$, $\overrightarrow{AB}$, $\overrightarrow{AC}$, $\overrightarrow{AD}$, and the new coordinate system is given by $M$, $\overrightarrow{MB}$, $\overrightarrow{MC}$, $\overrightarrow{MA}$. Find the coordinates of a point in the old coordinate system given its coordinates $x'$, $y'$, $z'$ in the new one.

**Test questions for final assessment in this section**

1. Operations om matrices and determinants.
2. Inverse matrix.
3. Systems of linear equations and their solution in matrix form.
4. Changing basis and coordinates.

#### K.5.2.3 Section 3

**Section title:** Lines in the plane and in the space

**Topics covered in this section:**

- General equation of a line in the plane
- General parametric equation of a line in the space
- Line as intersection between planes
- Vector equation of a line
- Distance from a point to a line
- Distance between lines
- Inter-positioning of lines

### What forms of evaluation were used to test students' performance in this section?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

### Typical questions for ongoing performance evaluation within this section

1. How to represent a line in the vector form?
2. What is the result of intersection of two planes in vector form?
3. How to derive the formula for the distance from a point to a line?
4. How to interpret geometrically the distance between lines?
5. List all possible inter-positions of lines in the space.

### Typical questions for seminar classes (labs) within this section

1. Two lines are given by the equations $\mathbf{r}\cdot\mathbf{n} = A$ and $\mathbf{r} = \mathbf{r}_0 + \mathbf{a}t$, and at that $\mathbf{a}\cdot\mathbf{n} \neq 0$. Find the position vector of the intersection point of these lines.
2. Find the distance from point $M_0$ with the position vector $\mathbf{r}_0$ to the line defined by the equation (a) $\mathbf{r} = \mathbf{r}_0 + \mathbf{a}t$; (b) $\mathbf{r}\cdot\mathbf{n} = A$.
3. Diagonals of a rhombus intersect at point $M(1;2)$, the longest of them being parallel to a horizontal axis. The side of the rhombus equals 2 and its obtuse angle is 120°. Compose the equations of the sides of this rhombus.
4. Compose the equations of lines passing through point $A(2;-4)$ and forming angles of 60° with the line $\frac{1-2x}{3} = \frac{3+2y}{-2}$.

### Test questions for final assessment in this section

1. Lines in the plane and in the space. Equations of lines.
2. Distance from a point to a line.
3. Distance between two parallel lines.
4. Distance between two skew lines.

#### K.5.2.4 Section 4

**Section title:** Planes in the space

**Topics covered in this section:**

- General equation of a plane
- Normalized linear equation of a plane
- Vector equation of a plane
- Parametric equation a plane
- Distance from a point to a plane
- Projection of a vector on the plane
- Inter-positioning of lines and planes
- Cross Product of two vectors
- Triple Scalar Product

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the difference between general and normalized forms of equations of a plane?
2. How to rewrite the equation of a plane in a vector form?
3. What is the normal to a plane?
4. How to interpret the cross products of two vectors?
5. What is the meaning of scalar triple product of three vectors?

**Typical questions for seminar classes (labs) within this section**

1. Find the cross product of (a) vectors $\mathbf{a}(3;-2;1)$ and $\mathbf{b}(2;-5;-3)$; (b) vectors $\mathbf{a}(3;-2;1)$ and $\mathbf{c}(-18;12;-6)$.
2. A triangle is constructed on vectors $\mathbf{a}(2;4;-1)$ and $\mathbf{b}(-2;1;1)$. (a) Find the area of this triangle. (b) Find the altitudes of this triangle.
3. Find the scalar triple product of $\mathbf{a}(1;2;-1)$, $\mathbf{b}(7;3;-5)$, $\mathbf{c}(3;4;-3)$.
4. It is known that basis vectors $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ have lengths of 1, 2, $2\sqrt{2}$ respectively, and $\angle(\mathbf{e}_1,\mathbf{e}_2) = 120°$, $\angle(\mathbf{e}_1,\mathbf{e}_3) = 135°$, $\angle(\mathbf{e}_2,\mathbf{e}_3) = 45°$. Find the volume of a parallelepiped constructed on vectors with coordinates $(-1;0;2)$, $(1;1\,4)$ and $(-2;1;1)$ in this basis.

**Test questions for final assessment in this section**

1. Planes in the space. Equations of planes.
2. Distance from a point to a plane, from a line to a plane.
3. Projection of a vector on the plane.

4. Cross product, its properties and geometrical interpretation.
5. Scalar triple product, its properties and geometrical interpretation.

#### K.5.2.5 Section 5

**Section title:** Quadratic curves

**Topics covered in this section:**

- Circle
- Ellipse
- Hyperbola
- Parabola
- Canonical equations
- Shifting of coordinate system
- Rotating of coordinate system
- Parametrization

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Formulate the canonical equation of the given quadratic curve.
2. Which orthogonal transformations of coordinates do you know?
3. How to perform a transformation of the coordinate system?
4. How to represent a curve in the space?

**Typical questions for seminar classes (labs) within this section**

1. Prove that a curve given by $34x^2 + 24xy + 41y^2 - 44x + 58y + 1 = 0$ is an ellipse. Find the major and minor axes of this ellipse, its eccentricity, coordinates of its center and foci. Find the equations of axes and directrices of this ellipse.
2. Determine types of curves given by the following equations. For each of the curves, find its canonical coordinate system (i.e. indicate the coordinates of origin and new basis vectors in the initial coordinate system) and its canonical equation. (a) $9x^2 - 16y^2 - 6x + 8y - 144 = 0$; (b) $9x^2 + 4y^2 + 6x - 4y - 2 = 0$; (c) $12x^2 - 12x - 32y - 29 = 0$; (d) $xy + 2x + y = 0$;
3. Find the equations of lines tangent to curve $6xy + 8y^2 - 12x - 26y + 11 = 0$ that are (a) parallel to line $6x + 17y - 4 = 0$; (b) perpendicular to line $41x - 24y + 3 = 0$; (c) parallel to line $y = 2$.

**Test questions for final assessment in this section**

1. Determine the type of a given curve with the use of the method of invariant.
2. Compose the canonical equation of a given curve.
3. Determine the canonical coordinate system for a given curve.

#### K.5.2.6 Section 6

**Section title:** Quadric surfaces

**Topics covered in this section:**

- General equation of the quadric surfaces
- Canonical equation of a sphere and ellipsoid
- Canonical equation of a hyperboloid and paraboloid
- Surfaces of revolution
- Canonical equation of a cone and cylinder
- Vector equations of some quadric surfaces

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**What forms of evaluation were used to test students' performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

1. What is the type of a quadric surface given by a certain equation?
2. How to compose the equation of a surface of revolution?
3. What is the difference between a directrix and generatrix?
4. How to represent a quadric surface in the vector form?

**Typical questions for seminar classes (labs) within this section**

1. For each value of parameter $a$ determine types of surfaces given by the equations: (a) $x^2+y^2-z^2=a$; (b) $x^2+a\left(y^2+z^2\right)=1$; (c) $x^2+ay^2=az$; (d) $x^2+ay^2=az+1$.
2. Find a vector equation of a right circular cone with apex $M_0\left(\mathbf{r}_0\right)$ and axis $\mathbf{r}=\mathbf{r}_0+\mathbf{a}t$ if it is known that generatrices of this cone form the angle of $\alpha$ with its axis.
3. Find the equation of a cylinder with radius $\sqrt{2}$ that has an axis $x=1+t$, $y=2+t$, $z=3+t$.
4. An ellipsoid is symmetric with respect to coordinate planes, passes through point $M(3;1;1)$ and circle $x^2+y^2+z^2=9$, $x-z=0$. Find the equation of this ellipsoid.

**Test questions for final assessment in this section**

1. Determine the type of a quadric surface given by a certain equation.
2. Compose the equation of a surface of revolution with the given directrix and generatrix.
3. Represent a given equation of a quadric surface in the vector form.

## K.6 Philosophy I (Logic)

- **Course name:** Philosophy I (Logic)
- **Course number:** BS-

### K.6.1 Course characteristics

#### K.6.1.1 Key concepts of the class

- Proofs
- Semantics of formulas
- Mappings

#### K.6.1.2 What is the purpose of this course?

Students need to be able to think abstractly. They need to know how to write simple mathematical proofs in English: in discrete mathematics, in analysis, in algebra… Proofs in English are mirrors of formal proofs; thus knowing a formalism for proofs can help students write mathematical proofs in English. But formal proofs might be used also for

- proving the correctness of programs;
- proving some properties of programs;
- specifying programs;
- producing correct programs.

#### K.6.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define

- Know some basic properties of mappings (injection, surjection, bijection…)
- Know some basic operations on mappings (composition, image…)
- Know some basic operations on sets (intersection, union, powerset, Cartesian products…)

- Know what Heyting algebras are
- Know what Boolean algebras are

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- The notion of model for propositional logic
- The notion of model for first-order logic
- Formal proofs in a natural deduction system

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply

- Write simple proofs in English
- Write simple formal proofs
- Use a proof-assistant to check a simple proof
- Compute the semantics of a formal formula.

#### K.6.1.4 Course evaluation

**Table K.16:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 10 |
| Interim performance assessment | 30 | 30 |
| Exams | 50 | 60 |

If necessary, please indicate freely your course's features in terms of students' performance assessment: If 40/100 * midterm grade + 50/100 * exam grade is more in favour of the student than 30/100*midterm grade + 60/100 * exam grade, then the final grade will be computed according to the former way instead of the latter way.

Table K.17: Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 60-79 |
| C. Satisfactory | 60-74 | 40-59 |
| D. Poor | 0-59 | 0-39 |

#### K.6.1.5 Grades range

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table M.73, but it may change slightly (usually reduced) depending on how the semester progresses.

#### K.6.1.6 Resources and reference material

- Handouts supplied by the instructor
-
-
-
-

### K.6.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

Table K.18: Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Set Theory | 20 |
| 2 | Proof Theory | 20 |
| 3 | Model Theory | 20 |

#### K.6.2.1 Section 1

**Section title:**

Set Theory

**Topics covered in this section:**

- Mappings
- Properties of mappings (injection, surjection, bijection...)
- Operations on mappings (restriction, direct and inevrse images...)
- Operations on sets (intersection, union, Cartesian product...)
- Cardinality (Hume's principle)

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Prove that any surjection is right-cancellative.
2. Prove that any injection is left-cancellative.
3. Show that the composite of two injections is an injection.
4. Show that every function with a right inverse is necessarily a surjection.

**Typical questions for seminar classes (labs) within this section**

1. Check if a given function is an injection.
2. Check if a given function is a surjection.
3. Show that every function with a left inverse is necessarily an injection.
4. Show that if $g \circ f$ is an injection, then $f$ is an injection.

**Test questions for final assessment in this section**

1. Show that if $g \circ f$ is a surjection, then $g$ is a surjection.
2. Show that inverse images preserve unions.
3. Show that, for any functions $f : X \to Y$ and $g : Y \to Z$, we have $\overline{\mathscr{P}}(g \circ f) = \overline{\mathscr{P}}(f) \circ \overline{\mathscr{P}}(g)$, where $\overline{\mathscr{P}}(f)$ and $\overline{\mathscr{P}}(g)$ are the inverse image operators of $f$ and $g$ respectively.

#### K.6.2.2 Section 2

**Section title:**

Proof Theory

**Topics covered in this section:**

- Formal proofs of propositional logic in natural deduction.
- Formal proofs of first-order logic in natural deduction.
- Use of a proof-assistant.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Given a partial proof on some proof-assistant, provide the current goal.
2. Give a formal proof in natural deduction of the judgement $\vdash ((A \Rightarrow \neg A) \Rightarrow \neg A)$ true.
3. Give a formal proof in natural deduction of the judgement $\vdash ((\neg A \Rightarrow A) \Rightarrow \neg\neg A)$ true.
4. Give a formal proof in natural deduction of the judgement $\vdash ((A \Rightarrow B) \Rightarrow (\neg B \Rightarrow \neg A))$ true.

**Typical questions for seminar classes (labs) within this section**

1. Give a formal proof in natural deduction of the judgement $\vdash P$ true, where $P$ is some given propositional formula.
2. Prove the judgement $\vdash ((X \Rightarrow (Y \Rightarrow Z)) \Rightarrow ((X \land Y) \Rightarrow Z))$ true on some proof-assistant.
3. Give a formal proof in natural deduction of the judgment $\vdash ((\exists y)(\forall x)P(x,y) \Rightarrow (\forall y)(\exists x)P(x,y))$ true

4. Prove on some proof-assistant that any injection is left-cancellative.

**Test questions for final assessment in this section**

1. Give a formal proof in natural deduction of the judgement $\vdash (((C \Rightarrow A) \vee (C \Rightarrow B)) \Rightarrow (C \Rightarrow (A \vee B)))$ true
2. Give a formal proof in natural deduction of the judgement $\vdash ((\forall x)(A(x) \wedge B(x)) \Rightarrow ((\forall x)A(x) \wedge (\forall x)B(x)))$ true.
3. Given a partial proof on some proof-assistant, complete it.

#### K.6.2.3 Section 3

**Section title:**

Model Theory

**Topics covered in this section:**

- Heyting Algebras
- Boolean Algebras
- Models of theories
- Completeness theorems
- Elementary equivalence of structures

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Compute the truth-value of a given propositional formula in a given Heyting algebra.
2. Recognize whether a given poset is a Heyting algebra.
3. Show that a given propositional formula is not provable.

### Typical questions for seminar classes (labs) within this section

1. Show that, in Heyting algebras, we necessarily have $\neg(a \vee b) = \neg a \wedge \neg b$.
2. Show that, in Heyting algebras, we do not necessarily have $\neg(a \wedge b) = \neg a \vee \neg b$.
3. Show that, in Boolean algebras, we necessarily have $\neg(a \wedge b) = \neg a \vee \neg b$.
4. Check whether a given first-order sentence is satisfied in a given structure.

### Test questions for final assessment in this section

1. Show that a given formula is not provable.
2. Show that a given formula is not provable without excluded middle.
3. Show that two given structures are elementary equivalent.
4. Show that two given structures are not elementary equivalent.

## K.7 Introduction to Programming II

- **Course name:** Introduction to Programming II
- **Course number:** XYZ
- **Knowledge area:** Programming Languages and Software Engineering

### K.7.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 1st year of BS
- **Semester of instruction:** 2nd semester
- **No. of Credits:** 4 ECTS
- **Total workload on average:** 144 hours overall
- **Class lecture hours:** 2 per week
- **Class tutorial hours:** 2 per week
- **Lab hours:** 2 per week
- **Individual lab hours:** 0
- **Frequency:** weekly throughout the semester
- **Grading mode:** letters: A, B, C, D

### K.7.2 Prerequisites

- Discrete Math/Logic
- Introduction to Programming I

### K.7.3 Course outline

Introduction to Programming II is the continuation of an introductory course to programming. It teaches in a more in-depth look at programming and at the development of software. The course is project oriented and it focusses on problem-solving and how to program well. Students will learn how to master the fundamental control structures, data structures, reasoning patterns and programming language mechanisms characterizing modern programming, as well as the fundamental rules of producing high-quality software. The course also introduces functional programming.

### K.7.4 Expected learning outcomes

After taking the course, students will

- master the fundamental rules of producing high-quality software.
- acquire in-depth programming background
- use higher-order functions to improve readability and efficiency of programs
- effectively decompose simple programming problems
- debug and test programs.

### K.7.5 Expected acquired core competences

- Object oriented programming
- Programming concepts and skills

### K.7.6 Textbook

- Bertrand Meyer: Touch of Class, Learning to Program Well with Objects and Contracts, Springer 2009, ISBN: 978-3-540-92144-8

### K.7.7 Reference material

Lecturing and lab slides and material will be provided

### K.7.8 Required computer resources

Students should have laptops with basic software for reading and editing document.

### K.7.9 Evaluation

- Course Project (50%)
- Final Exam (40%)
- Class and lab participation (10%)

## K.8 Data Structures and Algorithms

- **Course name:** Data Structures and Algorithms
- **Course number:** —

### K.8.1 Course characteristics

#### K.8.1.1 Key concepts of the class

- Algorithms
- Algorithm Analysis
- Algorithmic Strategies
- Data Structures

#### K.8.1.2 What is the purpose of this course?

This course provides an intensive treatment of a cross-section of the key elements of algorithms and data-structures, with an emphasis on implementing them in modern programming environments, and using them to solve real-world problems. The course will begin with the fundamentals of searching, sorting, lists, stacks, and queues, but will quickly build to cover more advanced topics, including trees, graphs, and algorithmic strategies. It will also cover the analysis of the performance and tractability of algorithms and will build on the concept of Abstract Data Types. A key focus of the course is on effective implementation and good design principles.

#### K.8.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?** By the end of the course, the students should be able to recognize and define

- Algorithms
- Abstract Data Types
- Data Structures
- Algorithmic Strategies
- Asymptotic Analysis
- Amortized Analysis

**- What should a student be able to understand at the end of the course?** By the end of the course, the students should be able to describe and explain (with examples)

- Difference between different abstract data types and data structures
- How to perform asymptotic and amortized analysis
- Difference between various algorithmic strategies
- Different algorithms: such as sorting, searching, etc.
- Different types of tree ADTs, their properties related algorithms
- Graphs, their properties, and related algorithms

**- What should a student be able to apply at the end of the course?** By the end of the course, the students should be able to apply

- Algorithmic strategies to solve real-life problems
- Asymptotic analysis to Analyze algorithms and software's complexity
- Trees and Graphs (and their theory) to solve complex problems

#### K.8.1.4 Course evaluation

**Table K.19:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 0 |
| Interim performance assessment | 30 | 30 |
| Exams | 50 | 70 |

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

#### K.8.1.5 Grades range

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table **??**, but it may change slightly (usually reduced) depending on how the semester progresses.

**Table K.20:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

#### K.8.1.6 Resources and reference material

- T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms. The MIT Press 2009.*
- *M. T. Goodrich, R. Tamassia, and M. H. Goldwasser. Data Structures and Algorithms in Java. WILEY 2014.*

### K.8.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.21:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Elementary Data Structures, Algorithmic Complexity and Approaches | 28 |
| 2 | Sorting Algorithms and Trees | 24 |
| 3 | Graphs | 28 |

#### K.8.2.1 Section 1

**Section title:** Elementary Data Structures, Algorithmic Complexity and Approaches

**Topics covered in this section:**

- Algorithms and Their Analysis
- Elementary Data Structures
- Hashing Map and Collision Handling
- Algorithmic Strategies

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. For a given function give an asymptotic upper bound using "big-Oh" notation
2. Compute the worst case running time of a given algorithm.
3. Insert items into a hashmap given a hash function and a collision handling scheme.
4. Given an algorithm, identify its algorithmic strategy

**Typical questions for seminar classes (labs) within this section**

1. How to implement various data structures?
2. Implement an algorithm for a given task having a desired worst case time complexity
3. Describe the difference between different types algorithmic strategies
4. Implement a hashmap
5. Solve various practical problems using different algorithmic strategies

**Test questions for final assessment in this section**

1. For a given function give an asymptotic upper bound using "big-Oh" notation
2. Compute the worst case running time of a given algorithm.
3. Insert items into a hashmap given a hash function and a collision handling scheme.
4. Given an algorithm, identify its algorithmic strategy

#### K.8.2.2 Section 2

**Section title:** Sorting Algorithms and Trees

**Topics covered in this section:**

- Comparison and Non-comparison Sort
- Binary Search Tree
- Balanced Binary Search Trees
- Tree Traversals
- Priority Queues and Binary Heaps

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Given a BST, answer different questions, such as (a) is the tree an AVL tree? What is the predecessor of a certain node? (b) Will after the removal of a certain node, the resulting tree will be a AVL tree or not?
2. Similar question as above but for other types of balanced binary search trees, including randomly built binary search trees.
3. Questions related to tree algorithms, such as tree traversals
4. Given a sorting problem defined under some constraints, what sorting algorithm will you use and why?

**Typical questions for seminar classes (labs) within this section**

1. Implement different types of binary search trees
2. Implement tree traversals
3. Implement different sorting algorithms, such as quicksort, countsort, bucket-sort, etc.

**Test questions for final assessment in this section**

1. Given an unbalanced AVL tree, perform double rotation and show the resulting tree.
2. Given a sequence of elements to be sorted, explain which sorting algorithm you would use to sort the input the fastest and why you chose this sorting algorithm.
3. Implement a sorting algorithm given a problem and specify the big-Oh running time for your algorithm.

#### K.8.2.3 Section 3

**Section title:** Graphs

**Topics covered in this section:**

- Graph Representations
- Searching in Graphs
- Minimum Spanning Tree
- Shortest Path
- Max-flow Min-cut

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Given a graph with a certain number of vertices and connected components, compute the largest number of edges that it might have?
2. What is the difference between adjacency list and adjacency matrix representation of a graph?

**Typical questions for seminar classes (labs) within this section**

1. Implement various graph representations
2. Given a computing problem, devise an algorithm to solve it using Graphs and then implement your algorithm.

**Test questions for final assessment in this section**

1. Give pseudocode for performing a certain operation in a required time complexity using the adjacency list representation.
2. Give pseudocode for performing a certain operation in a required time complexity using the adjacency list representation.
3. Calculate the maximum flow for a given flow network

## K.9 MathematicalAnalysis II

- **Course name:** Mathematical Analysis II
- **Course number:** BS-01

### K.9.1 Course Characteristics

#### K.9.1.1 Key concepts of the class

- Multivariate calculus: derivatives, differentials, maxima and minima
- Multivariate integration
- Functional series. Fourier series
- Integrals with parameters

#### K.9.1.2 What is the purpose of this course?

The goal of the course is to study basic mathematical concepts that will be required in further studies. The course is based on Mathematical Analysis I, and the concepts studied there are widely used in this course. The course covers differentiation and integration of functions of several variables. Some more advanced concepts, as uniform convergence of series and integrals, are also considered, since they are important for understanding applicability of many theorems of mathematical analysis. In the end of the course some useful applications are covered, such as gamma-function, beta-function, and Fourier transform.

#### K.9.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to:

- find partial and directional derivatives of functions of several variables;
- find maxima and minima for a function of several variables
- use Fubiniâ€™s theorem for calculating multiple integrals
- calculate line and path integrals

- distinguish between point wise and uniform convergence of series and improper integrals
- decompose a function into Fourier series
- calculate Fourier transform of a function

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to understand:

- how to find minima and maxima of a function subject to a constraint
- how to represent double integrals as iterated integrals and vice versa
- what the length of a curve and the area of a surface is
- properties of uniformly convergent series and improper integrals
- beta-function, gamma-function and their properties
- how to find Fourier transform of a function

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

- find multiple, path, surface integrals
- find the range of a function in a given domain
- decompose a function into Fourier series

#### K.9.1.4 Course evaluation

**Table K.22:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Test 1 | ? | 10 |
| Midterm | ? | 25 |
| Test 2 | ? | 10 |
| Participation | ? | 5 |
| Final exam | ? | 50 |

#### K.9.1.5 Grades range

**Table K.23:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 85-100 |
| B. Good | 75-89 | 65-84 |
| C. Satisfactory | 60-74 | 45-64 |
| D. Poor | 0-59 | 0-44 |

#### K.9.1.6 Resources and reference material

- Robert A. Adams, Christopher Essex (2017) Calculus. A Complete Course, Pearson
- Jerrold Marsden, Alan Weinstein (1985) Calculus (in three volumes; volumes 2 and 3), Springer

### K.9.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.24:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Differential Analysis of Functions of Several Variables | 24 |
| 2 | Integration of Functions of Several Variables | 30 |
| 3 | Uniform Convergence of Functional Series. Fourier Series | 18 |
| 4 hline | Integrals with Parameter(s) | 18 |

#### K.9.2.1 Section 1

**Section title:** Differential Analysis of Functions of Several Variables

**Topics covered in this section:**

- Limits of functions of several variables
- Partial and directional derivatives of functions of several variables. Gradient
- Differentials of functions of several variables. Taylor formula
- Maxima and minima for functions of several variables
- Maxima and minima for functions of several variables subject to a constraint

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Find $\lim\limits_{x\to 0}\lim\limits_{y\to 0} u(x;y)$, $\lim\limits_{y\to 0}\lim\limits_{x\to 0} u(x;y)$ and $\lim\limits_{(x;y)\to(0;0)} u(x;y)$ if $u(x;y)=\frac{x^2y+xy^2}{x^2-xy+y^2}$.
2. Find the differential of a function: (a) $u(x;y)=\ln\left(x+\sqrt{x^2+y^2}\right)$; (b) $u(x;y)=\ln\sin\frac{x+1}{\sqrt{y}}$.
3. Find the differential of $u(x;y)$ given implicitly by an equation $x^3+2y^3+u^3-3xyu+2y-3=0$ at points $M(1;1;1)$ and $N(1;1;-2)$.
4. Find maxima and minima of a function subject to a constraint (or several constraints):
   (a) $u=x^2y^3z^4, \quad 2x+3y+4z=18,\ x>0,\ y>0,\ z>0;$
   (b) $u=x-y+2z, \quad x^2+y^2+2z^2=16;$
   (c) $u=\sum\limits_{i=1}^{k} a_ix_i^2, \quad \sum\limits_{i=1}^{k} x_i=1,\ a_i>0;$

**Typical questions for seminar classes (labs) within this section**

1. Let us consider $u(x;y)=\begin{cases}1, & x=y^2,\\ 0, & x\neq y^2.\end{cases}$ Show that this function has a limit at the origin along any straight line that passes through it (and all these limits are equal to each other), yet this function does not have limit as $(x;y)\to(0;0)$.

2. Find the largest possible value of directional derivative at point $M(1;-2;-3)$ of function $f = \ln xyz$.
3. Find maxima and minima of functions $u(x, y)$ given implicitly by the equations:
   (a) $x^2 + y^2 + u^2 - 4x - 6y - 4u + 8 = 0,\ u > 2$;
   (b) $x^3 - y^2 + u^2 - 3x + 4y + u - 8 = 0$.
4. Find maxima and minima of functions subject to constraints:
   (a) $u = xy^2,\ x + 2y - 1 = 0$;
   (b) $u = xy + yz,\ x^2 + y^2 = 2,\ y + z = 2,\ y > 0$.

**Test questions for final assessment in this section**

1. Find all points where the differential of a function $f(x;y) = (5x+7y-25)e^{-x^2-xy-y^2}$ is equal to zero.
2. Show that function $\varphi = f\left(\frac{x}{y}; x^2 + y - z^2\right)$ satisfies the equation $2xz\varphi_x + 2yz\varphi_y + \left(2x^2 + y\right)\varphi_z = 0$.
3. Find maxima and minima of function $u = 2x^2 + 12xy + y^2$ under condition that $x^2 + 4y^2 = 25$. Find the maximum and minimum value of a function
4. $u = \left(y^2 - x^2\right)e^{1-x^2+y^2}$ on a domain given by inequality $x^2 + y^2 \leq 4$;

#### K.9.2.2 Section 2

**Section title:** Integration of Functions of Several Variables

**Topics covered in this section:**

- Z-test
- Double integrals. Fubiniâ€™s theorem and iterated integrals
- Substituting variables in double integrals. Polar coordinates
- Triple integrals. Use of Fubiniâ€™s theorem
- Spherical and cylindrical coordinates
- Path integrals
- Area of a surface
- Surface integrals

### What forms of evaluation were used to test students' performance in this section?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

### Typical questions for ongoing performance evaluation within this section

1. Represent double integrals below as an iterated integrals (or a sum of iterated integrals) with different orders of integration: $\iint\limits_D f(x;y)\,dx\,dy$ where $D = \{(x;y)\,|\,x^2+y^2 \le 9,\ x^2+(y+4)^2 \ge 25\}$.
2. Represent integral $I = \iiint\limits_D f(x;y;z)\,dx\,dy\,dz$ as iterated integrals with all possible (i.e. 6) orders of integration; $D$ is bounded by $x=0$, $x=a$, $y=0$, $y=\sqrt{ax}$, $z=0$, $z=x+y$.
3. Find line integrals of a scalar fields $\int\limits_\Gamma (x+y)\,ds$ where $\Gamma$ is boundary of a triangle with vertices $(0;0)$, $(1;0)$ and $(0;1)$.

### Typical questions for seminar classes (labs) within this section

1. Change order of integration in the iterated integral $\int\limits_0^{\sqrt{2}} dy \int\limits_y^{\sqrt{4-y^2}} f(x;y)\,dx$.
2. Find the volume of a solid given by $0 \le z \le x^2$, $x+y \le 5$, $x-2y \ge 2$, $y \ge 0$.
3. Change into polar coordinates and rewrite the integral as a single integral: $\iint\limits_G f\left(\sqrt{x^2+y^2}\right)dx\,dy$, $G = \{(x;y)\,|\,x^2+y^2 \le x;\ x^2+y^2 \le y\}$.
4. Having ascertained that integrand is an exact differential, calculate the integral along a piecewise smooth plain curve that starts at $A$ and finishes at $B$: $\int\limits_\Gamma \left(x^4+4xy^3\right)dx + \left(6x^2y^2-5y^4\right)dy$, $A(-2;-1)$, $B(0;3)$;

**Test questions for final assessment in this section**

1. Domain $G$ is bounded by lines $y = 2x$, $y = x$ and $y = 2$. Rewrite integral $\iint_G f(x)\,dx\,dy$ as a single integral.
2. Represent the integral $\iint_G f(x;y)\,dx\,dy$ as iterated integrals with different order of integration in polar coordinates if $G = \{(x;y)\,|\,a^2 \le x^2+y^2 \le 4a^2;\ |x|-y \ge 0\}$.
3. Find the integral making an appropriate substitution: $\iiint_G (x^2-y^2)(z+x^2-y^2)\,dx\,dy\,dz$, $G = \{(x;y;z)\,|\,x-1<y<x;\ 1-x<y<2-x;\ 1-x^2+y^2<z<y^2-x^2+2x\}$.
4. Use divergence theorem to find the following integrals $\iint_S (1+2x)\,dy\,dz + (2x+3y)\,dz\,dx + (3y+4z)\,dx\,dy$ where $S$ is the outer surface of a tetrahedron $\frac{x}{a}+\frac{y}{b}+\frac{z}{c} \le 1$, $x \ge 0$, $y \ge 0$, $z \ge 0$;

### K.9.2.3 Section 3

**Section title:** Uniform Convergence of Functional Series. Fourier Series

**Topics covered in this section:**

- Uniform and point wise convergence of functional series
- Properties of uniformly convergent series
- Fourier series. Sufficient conditions of convergence and uniform convergence
- Besselâ€™s inequality and Parsevalâ€™s identity.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Find out whether the following functional series converges uniformly on the indicated intervals. Justify your answer. $\sum_{n=1}^{\infty} e^{-n(x^2+2\sin x)}$, $\Delta_1 = (0;1]$, $\Delta_2 = [1;+\infty)$;

2. $\sum\limits_{n=1}^{\infty} \frac{\sqrt{nx^3}}{x^2+n^2}$, $\Delta_1 = (0;1)$, $\Delta_2 = (1;+\infty)$
3. Show that sequence $f_n(x) = nx(1-x)^n$ converges non-uniformly on $[0;1]$ to a continuous function $f(x)$, but $\lim\limits_{n\to+\infty} \int\limits_0^1 f_n(x)\,dx = \lim\limits_{n\to+\infty} \int\limits_0^1 f(x)\,dx$.
4. Decompose the following function determined on $[-\pi;\pi]$ into Fourier series using the standard trigonometric system $\{1; \cos kx; \sin kx\}|_{k=1}^{\infty}$. Draw the graph of the sum of Fourier series obtained. $f(x) = \begin{cases} 1, \ 0 \le x \le \pi, \\ 0, \ -\pi \le x < 0. \end{cases}$
5. Prove that if for an absolutely integrable function $f(x)$ on $[-\pi;\pi]$
   (a) $f(x+\pi) = f(x)$ then $a_{2k-1} = b_{2k-1} = 0$, $k \in \mathbb{N}$;
   (b) $f(x+\pi) = -f(x)$ then $a_0 = 0$, $a_{2k} = b_{2k} = 0$, $k \in \mathbb{N}$.

**Typical questions for seminar classes (labs) within this section**

1. Show that sequence $f_n(x) = nx\left(1-x^2\right)^n$ converges on $[0;1]$ to a continuous function $f(x)$, and at that $\lim\limits_{n\to+\infty} \int\limits_0^1 f_n(x)\,dx \ne \lim\limits_{n\to+\infty} \int\limits_0^1 f(x)\,dx$.
2. Show that sequence $f_n(x) = x^3 + \frac{1}{n}\sin\left(nx + \frac{n\pi}{2}\right)$ converges uniformly on $\mathbb{R}$, but $\left(\lim\limits_{n\to+\infty} f_n(x)\right)' \ne \lim\limits_{n\to+\infty} f_n'(x)$.
3. Decompose $\cos\alpha x$, $\alpha \notin \mathbb{Z}$ into Fourier series on $[-\pi;\pi]$. Using this decomposition prove that $\cot y = \frac{1}{y} + \sum\limits_{k=1}^{\infty} \frac{2y}{y^2-\pi^2k^2}$.
4. Function $f(x)$ is absolutely integrable on $[0;\pi]$, and $f(\pi - x) = f(x)$. Prove that
   (a) if it is decomposed into Fourier series of sines then $b_{2k} = 0$, $k \in \mathbb{N}$;
   (b) if it is decomposed into Fourier series of cosines then $a_{2k-1} = 0$, $k \in \mathbb{N}$.
5. (a) Decompose $f(x) = \begin{cases} 1, \ |x| < \alpha, \\ 0, \ \alpha \leqslant |x| < \pi \end{cases}$ into Fourier series using the standard trigonometric system.
   (b) Using Parseval's identity find $\sigma_1 = \sum\limits_{k=1}^{\infty} \frac{\sin^2 k\alpha}{k^2}$ and $\sigma_2 = \sum\limits_{k=1}^{\infty} \frac{\cos^2 k\alpha}{k^2}$.

### K.9.2.4 Test questions for final assessment in the course

1. Find out whether the following functional series converge uniformly on the indicated intervals. Justify your answer. $\sum\limits_{n=1}^{\infty} \frac{xn+\sqrt{n}}{n+x} \ln\left(1 + \frac{x}{n\sqrt{n}}\right)$, $\Delta_1 = (0;1)$, $\Delta_2 = (1;+\infty)$;

2. Show that sequence $f_n(x) = \frac{\sin nx}{\sqrt{n}}$ converges uniformly on $\mathbb{R}$ to a differentiable function $f(x)$, and at that $\lim\limits_{n\to+\infty} f_n'(0) \neq f'(0)$.

### K.9.2.5 Section 1

**Section title:** Integrals with Parameter(s)

**Topics covered in this section:**

- Definite integrals with parameters
- Improper integrals with parameters. Uniform convergence
- Properties of uniformly convergent integrals
- Beta-function and gamma-function
- Fourier transform

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Find out if $\int\limits_0^1 \left( \lim\limits_{\alpha\to 0} \frac{2x\alpha^2}{\left(\alpha^2+x^2\right)^2} \right) dx = \lim\limits_{\alpha\to 0} \int\limits_0^1 \frac{2x\alpha^2}{\left(\alpha^2+x^2\right)^2}\, dx.$
2. Differentiating the integrals with respect to parameter $\varphi$, find it: $I(\alpha) = \int\limits_0^{\pi/2} \ln\left(\alpha^2 - \sin^2\varphi\right) d\varphi$, $\alpha > 1$.
3. Prove that the following integral converges uniformly on the indicated set. $\int\limits_0^{+\infty} e^{-\alpha x}\cos 2x\, dx$, $\Delta = [1; +\infty)$;
4. It is known that Dirichlet's integral $\int\limits_0^{+\infty} \frac{\sin x}{x}\, dx$ is equal to $\frac{\pi}{2}$. Find the values of the following integrals using Dirichlet's integral

(a) $\int\limits_0^{+\infty} \frac{\sin}{\alpha x} x\,dx,\ \alpha \neq 0;$

(b) $\int\limits_0^{+\infty} \frac{\sin x - x\cos x}{x^3}\,dx.$

**Typical questions for seminar classes (labs) within this section**

1. Find out if $\int\limits_0^1 \left( \int\limits_0^1 f(x,\alpha)\,d\alpha \right) dx = \int\limits_0^1 \left( \int\limits_0^1 f(x,\alpha)\,dx \right) d\alpha$ if $f(x;\alpha) = \frac{\alpha - x}{(\alpha + x)^3}$.
2. Find $\Phi'(\alpha)$ if $\Phi(\alpha) = \int\limits_1^2 \frac{e^{\alpha x^2}}{x}\,dx$.
3. Differentiating the integral with respect to parameter $\alpha$, find it: $I(\alpha) = \int\limits_0^{\pi} \frac{1}{\cos x} \ln \frac{1+\alpha\cos x}{1-\alpha\cos x}\,dx$, $|\alpha| < 1$.
4. Find Fourier transform of the following functions:

   (a) $f(x) = \begin{cases} 1, & |x| \le 1, \\ 0, & |x| > 1; \end{cases}$

5. Let $\widehat{f}(y)$ be Fourier transform of $f(x)$. Prove that Fourier transform of $e^{i\alpha x} f(x)$ is equal to $\widehat{f}(y - \alpha)$, $\alpha \in \mathbb{R}$.

**Test questions for final assessment in this section**

1. Find out if $\int\limits_0^1 \left( \int\limits_0^1 f(x,\alpha)\,d\alpha \right) dx = \int\limits_0^1 \left( \int\limits_0^1 f(x,\alpha)\,dx \right) d\alpha$ if $f(x;\alpha) = \frac{\alpha^2 - x^2}{(\alpha^2 + x^2)^2}$.
2. Find $\Phi'(\alpha)$ if $\Phi(\alpha) = \int\limits_0^{\alpha} \frac{\ln(1+\alpha x)}{x}\,dx$.
3. Prove that the following integral converges uniformly on the indicated set. $\int\limits_{-\infty}^{+\infty} \frac{\cos\alpha x}{4+x^2}\,dx$, $\Delta = \mathbb{R}$;
4. Find Fourier integral for $f(x) = \begin{cases} 1, & |x| \le \tau, \\ 0, & |x| > \tau; \end{cases}$

# K.10 Analytical Geometry & Linear Algebra – II

- **Course name:** Analytical Geometry & Linear Algebra – II
- **Course number:** XYZ
- **Subject area:** Math

## K.10.1 Course characteristics

### K.10.1.1 Key concepts of the class

- fundamental principles of linear algebra,
- concepts of linear algebra objects and their representation in vector-matrix form

### K.10.1.2 What is the purpose of this course?

This course covers matrix theory and linear algebra, emphasizing topics useful in other disciplines. Linear algebra is a branch of mathematics that studies systems of linear equations and the properties of matrices. The concepts of linear algebra are extremely useful in physics, data sciences, and robotics. Due to its broad range of applications, linear algebra is one of the most widely used subjects in mathematics.

### K.10.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

- List basic notions of linear algebra
- Understand key principles involved in solution of linear equation systems and the properties of matrices
- Linear regression analysis
- Fast Fourier Transform
- How to find eigenvalues and eigenvectors for matrix diagonalization and single value decomposition

**- What should a student be able to understand at the end of the course?**

- Key principles involved in solution of linear equation systems and the properties of matrices
- Become familiar with the four fundamental subspaces
- Linear regression analysis
- Fast Fourier Transform
- How to find eigenvalues and eigenvectors for matrix diagonalization and single value decomposition

**- What should a student be able to apply at the end of the course?**

- Linear equation system solving by using the vector-matrix approach
- Make linear regression analysis
- Fast Fourier Transform
- To find eigenvalues and eigenvectors for matrix diagonalization and single value decomposition

#### K.10.1.4 Course evaluation

**Table K.25:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 20 |
| Interim performance assessment | 30 | 30 |
| Exams | 50 | 50 |

#### K.10.1.5 Grades range

#### K.10.1.6 Resources and reference material

**Textbooks:**

- Gilbert Strang. Linear Algebra and Its Applications, 4th Edition, Brooks Cole, 2006. ISBN: 9780030105678
- Gilbert Strang. Introduction to Linear Algebra, 4th Edition, Wellesley, MA: Wellesley-Cambridge Press, 2009. ISBN: 9780980232714

**Table K.26:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 85-100 |
| B. Good | 75-89 | 65-84 |
| C. Satisfactory | 60-74 | 50-64 |
| D. Poor | 0-59 | 0-49 |

**Reference material:**

- Gilbert Strang, Brett Coonley, Andrew Bulman-Fleming. Student Solutions Manual for Strang's Linear Algebra and Its Applications, 4th Edition, Thomson Brooks, 2005. ISBN-13: 9780495013259

## K.10.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.27:** Course Sections

| Section Number | Section Title | Lectures (hours) | Seminars (labs) | Self-study | Knowledge evaluation |
|---|---|---|---|---|---|
| 1 | Linear equation system solving by using the vector-matrix approach. | 16 | 8 | 24 | 2 |
| 2 | Linear regression analysis and decomposition A=QR. | 12 | 6 | 18 | 2 |
| 3 | Fast Fourier Transform. Matrix Diagonalization. | 12 | 6 | 18 | 2 |
| 4 | Symmetric, positive definite and similar matrices. Singular value decomposition. | 12 | 6 | 18 | 2 |
| Final examination | | | | | 2 |

### K.10.2.1 Section 1

**Section title:** Linear equation system solving by using the vector-matrix approach

**Topics covered in this section:**

- The geometry of linear equations. Elimination with matrices.
- Matrix operations, including inverses. *LU* and *LDU* factorization.
- Transposes and permutations. Vector spaces and subspaces.

- The null space: Solving $Ax = 0$ and $Ax = b$. Row reduced echelon form. Matrix rank.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How to perform Gauss elimination?
2. How to perform matrices multiplication?
3. How to perform LU factorization?
4. How to find complete solution for any linear equation system Ax=b?

**Typical questions for seminar classes (labs) within this section**

1. Find the solution for the given linear equation system $Ax = b$ by using Gauss elimination.
2. Perform $A = LU$ factorization for the given matrix $A$.
3. Factor the given symmetric matrix $A$ into $A = LDL^T$ with the diagonal pivot matrix $D$.
4. Find inverse matrix $A^-1$ for the given matrix $A$.

**Test questions for final assessment in this section**

1. Find linear independent vectors (exclude dependent): $\overrightarrow{a} = [4,0,3,2]^T$, $\overrightarrow{b} = [1,-7,4,5]^T$, $\overrightarrow{c} = [7,1,5,3]^T$, $\overrightarrow{d} = [-5,-3,-3,-1]^T$, $\overrightarrow{e} = [1,-5,2,3]^T$. Find $rank(A)$ if $A$ is a composition of this vectors. Find $rank(A^T)$.
2. Find $E$: $EA = U$ ($U$ – upper-triangular matrix). Find $L = E^-1$, if $A = \begin{pmatrix} 2 & 5 & 7 \\ 6 & 4 & 9 \\ 4 & 1 & 8 \end{pmatrix}$.

3. Find complete solution for the system $Ax = b$, if $b = [7, 18, 5]^T$ and $A = \begin{pmatrix} 6 & -2 & 1 & -4 \\ 4 & 2 & 14 & -31 \\ 2 & -1 & 3 & -7 \end{pmatrix}$.

   Provide an example of vector b that makes this system unsolvable.

#### K.10.2.2 Section 2

**Section title:** Linear regression analysis and decomposition $A = QR$.

**Topics covered in this section:**

- Independence, basis and dimension. The four fundamental subspaces.
- Orthogonal vectors and subspaces. Projections onto subspaces
- Projection matrices. Least squares approximations. Gram-Schmidt and A = QR.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is linear independence of vectors?
2. Define the four fundamental subspaces of a matrix?
3. How to define orthogonal vectors and subspaces?
4. How to define orthogonal complements of the space?
5. How to find vector projection on a subspace?
6. How to perform linear regression for the given measurements?
7. How to find an orthonormal basis for the subspace spanned by the given vectors?

**Typical questions for seminar classes (labs) within this section**

1. Check out linear independence of the given vectors
2. Find four fundamental subspaces of the given matrix.
3. Check out orthogonality of the given subspaces.
4. Find orthogonal complement for the given subspace.
5. Find vector projection on the given subspace.
6. Perform linear regression for the given measurements.
7. Find an orthonormal basis for the subspace spanned by the given vectors.

**Test questions for final assessment in this section**

1. Find the dimensions of the four fundamental subspaces associated with $A$, depending on the parameters $a$ and $b$: $A = \begin{pmatrix} 7 & 8 & 5 & 3 \\ 4 & a & 3 & 2 \\ 6 & 8 & 4 & b \\ 3 & 4 & 2 & 1 \end{pmatrix}$.
2. Find a vector $x$ orthogonal to the Row space of matrix $A$, and a vector $y$ orthogonal to the $C(A)$, and a vector $z$ orthogonal to the $N(A)$: $A = \begin{pmatrix} 1 & 2 & 2 \\ 3 & 4 & 2 \\ 4 & 6 & 4 \end{pmatrix}$.
3. Find the best straight-line $y(x)$ fit to the measurements: $y(-2) = 4$, $y(-1) = 3$, $y(0) = 2$, $y(1) - 0$.
4. Find the projection matrix $P$ of vector $[4,3,2,0]^T$ onto the $C(A)$: $A = \begin{pmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$.
5. Find an orthonormal basis for the subspace spanned by the vectors: $\overrightarrow{a} = [-2,2,0,0]^T$, $\overrightarrow{b} = [0,1,-1,0]^T$, $\overrightarrow{c} = [0,1,0,-1]^T$. Then express $A = [a,b,c]$ in the form of $A = QR$

#### K.10.2.3 Section 3

**Section title:** Fast Fourier Transform. Matrix Diagonalization.

**Topics covered in this section:**

- Complex Numbers. Hermitian and Unitary Matrices.
- Fourier Series. The Fast Fourier Transform
- Eigenvalues and eigenvectors. Matrix diagonalization.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Make the definition of Hermitian Matrix.
2. Make the definition of Unitary Matrix.
3. How to find matrix for the Fourier transform?
4. When we can make fast Fourier transform?
5. How to find eigenvalues and eigenvectors of a matrix?
6. How to diagonalize a square matrix?

**Typical questions for seminar classes (labs) within this section**

1. Check out is the given matrix Hermitian.
2. Check out is the given matrix Unitary.
3. Find the matrix for the given Fourier transform.
4. Find eigenvalues and eigenvectors for the given matrix.
5. Find diagonalize form for the given matrix.

**Test questions for final assessment in this section**

1. Find eigenvector of the circulant matrix $C$ for the eigenvalue = $c_1+c_2+c_3+c_4$:
$C = \begin{pmatrix} c_1 & c_2 & c_3 & c_4 \\ c_4 & c_1 & c_2 & c_3 \\ c_3 & c_4 & c_1 & c_2 \\ c_2 & c_3 & c_4 & c_1 \end{pmatrix}$.
2. Diagonalize this matrix: $A = \begin{pmatrix} 2 & 1-i \\ 1+i & 3 \end{pmatrix}$.
3. $A$ is the matrix with full set of orthonormal eigenvectors. Prove that $AA = A^H A^H$.
4. Find all eigenvalues and eigenvectors of the cyclic permutation matrix $P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$.

#### K.10.2.4 Section 4

**Section title:** Symmetric, positive definite and similar matrices. Singular value decomposition.

**Topics covered in this section:**

- Linear differential equations.
- Symmetric matrices. Positive definite matrices.
- Similar matrices. Left and right inverses, pseudoinverse. Singular value decomposition (SVD).

**What forms of evaluation were used to test students' performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

1. How to solve linear differential equations?
2. Make the definition of symmetric matrix?
3. Make the definition of positive definite matrix?
4. Make the definition of similar matrices?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

5. How to find left and right inverses matrices, pseudoinverse matrix?
6. How to make singular value decomposition of the matrix?

**Typical questions for seminar classes (labs) within this section**

1. Find solution of the linear differential equation.
2. Make the definition of symmetric matrix.
3. Check out the given matrix on positive definess
4. Check out the given matrices on similarity.
5. For the given matrix find left and right inverse matrices, pseudoinverse matrix.
6. Make the singular value decomposition of the given matrix.

**Test questions for final assessment in this section**

1. Find $det(e^A)$ for $A = \begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix}$.
2. Write down the first order equation system for the following differential equation and solve it:
   $d^3y/dx + d^2y/dx - 2dy/dx = 0$
   $y"(0) = 6,\ y'(0) = 0,\ y(0) = 3.$
   Is the solution of this system will be stable?
3. For which $a$ and $b$ quadratic form $Q(x, y, z)$ is positive definite:
   $Q(x, y, z) = ax^2 + y^2 + 2z^2 + 2bxy + 4xz$
4. Find the SVD and the pseudoinverse of the matrix $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$.

## K.11 Theoretical Computer Science

- **Course name:** Theoretical Computer Science
- **Course number:** BS-

### K.11.1 Course characteristics

#### K.11.1.1 Key concepts of the class

- Automata Theory
- Formal Grammars
- Computability

#### K.11.1.2 What is the purpose of this course?

A good software developer ignorant of how the mechanics of a compiler works is not better than a good pilot when it comes to fix the engine and he will definitively not be able to provide more than average solutions to the problems he is employed to solve. Like automotive engineering teach us, races can only be won by the right synergy of a good driving style and mechanics. Most importantly, limits of computation cannot be ignored in the same way we precisely know how accelerations, forces and frictions prevent us from racing at an unlimited speed. This course will investigate the prerequisites to understand compilers functioning. Although the act of compilation appears deceptively simple to most of the modern developers, great minds and results are behind the major achievements that made this possible. All starts with the Epimenides paradox (about 600 BC), which emphasizes a problem of self-reference in logic and brings us to the short time window between WWI and WW2 when, in 1936, Alan Turing proved that a general procedure to identify algorithm termination simply does not exist. Another major milestone has been reached by Noam Chomsky in 1956 with his description of a hierarchy of grammars. In this long historical timeframe we can put most of the bricks with which we build modern compilers. The course will be an historical tour through the lives of some of the greatest minds who ever lived on this planet.

#### K.11.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define

- Define a formal language
- List different computational models
- Define computational models such as Finite State Automata and Pushdown Automata
- List different types of Formal Grammars
- Define computability and related concepts
- List applications for automata theory and formal grammar

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- Describe the basic mathematical machinery behind automata theory and how can be applied to programming languages compilers
- Explain Strengths and weaknesses of specific computational model
- Explain Finite State Automata and Pushdown Automata
- Abstract systems using the given models

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply

- Formally modelling a system
- Reasoning about verification of program properties
- Specifying a system as Finite State Automata, Pushdown Automata or Turing Machine
- Coding in programming languages an emulator of Finite State Automata
- Using proof techniques by diagonalization

**Table K.28:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 40 |
| Interim performance assessment | 30 | 30 |
| Exams | 50 | 30 |

#### K.11.1.4 Course evaluation

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

#### K.11.1.5 Grades range

**Table K.29:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 65-79 |
| C. Satisfactory | 60-74 | 50-64 |
| D. Poor | 0-59 | 0-49 |

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table M.73, but it may change slightly (usually reduced) depending on how the semester progresses.

#### K.11.1.6 Resources and reference material

- Handouts supplied by the instructor
-
-
-

### K.11.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.30:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Automata Theory | 42 |
| 2 | Formal Grammars | 24 |
| 3 | Computability | 24 |

#### K.11.2.1 Section 1

**Section title:**

Automata Theory

**Topics covered in this section:**

- Languages
- Finite State Automata
- Pushdown Automata
- Nondeterminism
- Turing Machines

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is a Finite State Automaton?
2. What is a Pushdown Automaton?
3. What is a Turing Machine?
4. What is a nondeterministic automaton?
5. Given a specific language define a corresponding Finite State Automaton
6. Given a specific language define a corresponding Pushdown Automaton
7. State the difference between Finite State Automata and Pushdown Automata
8. Computing the intersection, union, complement of two automata
9. What is Pumping Lemma? Example of applications.

10. Operations on Automata

**Typical questions for seminar classes (labs) within this section**

1. Check if a given language is recognized by a specific Finite State Automaton
2. Prove with Pumping Lemma that a language is regular
3. Check if a given language is recognized by a specific Pushdown Automaton
4. Check if a given language is recognized by a specific Turing Machine
5. Define a correct automaton given a specific language

**Test questions for final assessment in this section**

1. Check if a given language is recognized by a specific Finite State Automaton
2. Prove with Pumping Lemma that a language is regular.
3. Check if a given language is recognized by a specific Pushdown Automaton
4. Check if a given language is recognized by a specific Turing Machine
5. Define a correct automaton given a specific language

#### K.11.2.2 Section 2

**Section title:**

Formal Grammars

**Topics covered in this section:**

- Chomsky Hierarchy
- Regular Expressions
- Relationships with Automata

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is Chomsky Hierarchy?
2. What is a Regular language?
3. What is a Context-free language?
4. What is a Context-sensitive language?
5. What is a regular expression?

**Typical questions for seminar classes (labs) within this section**

1. Given a specific Finite State Automaton define a grammar for the corresponding language
2. Given a specific Pushdown Automaton define a grammar for the corresponding language
3. Given a regular expression design the corresponding Finite state Automaton
4. Given a Finite state Automaton define the regular expression for the corresponding language

**Test questions for final assessment in this section**

1. What is Chomsky Hierarchy?
2. What is a Regular language?
3. Given a regular expression design the corresponding Finite state Automaton
4. Given a specific Pushdown Automaton define a grammar for the corresponding language

#### K.11.2.3 Section 3

**Section title:**

Computability

**Topics covered in this section:**

- Undecidability
- Halting Problem
- Rice Theorem

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is Halting Problem?
2. What is Rice theorem?
3. What is an undecidable problem
4. What is a Turing Machine?
5. What is Goedelization?

**Typical questions for seminar classes (labs) within this section**

1. Given a specific computational problem showing that it is undecidable (via reduction to a known problem, for example)
2. Show a proof via diagonalization of Halting Problem
3. Show a proof via diagonalization of Rice Theorem
4. Give an example of an undecidable problem
5. Show that a Turing Machine with multiple tapes is equivalent to a Turing Machine with a single tape

**Test questions for final assessment in this section**

1. What is a Turing Machine?
2. What is Halting Problem?
3. What is Rice theorem?
4. Show a proof via diagonalization of Halting Problem
5. Show a proof via diagonalization of Rice Theorem
6. Show that a Turing Machine with multiple tapes is equivalent to a Turing Machine with a single tape

## K.12 Probability and Statistics

- **Course name:** Probability and Statistics
- **Course number:** XYZ
- **Knowledge area:** Math, Computational Science

### K.12.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 2nd year of BS
- **Semester of instruction:** 2nd semester
- **No. of Credits:** 4 ECTS
- **Total workload on average:** 144 hours overall
- **Class lecture hours:** 2 per week
- **Class tutorial hours:** 2 per week
- **Lab hours:** 2 per week
- **Individual lab hours:** 0
- **Frequency:** weekly throughout the semester
- **Grading mode:** letters: A, B, C, D

### K.12.2 Prerequisites

- Mathematical Analysis I
- Discrete Math and Logic

### K.12.3 Course outline

The course is designed to provide Software Engineers and Computer Scientists by correct knowledge of basic (core) concepts, definitions, theoretical results and applied methods & techniques of Probability Theory and Mathematical Statistics. All definitions and theorem statements (that will be given in lectures and that are needed to explain the key-words listed above) will be formal, but just few of these theorems will be proved formally. Instead (in lab or problem-solving classes) we will try these definitions and theorems on work with routine exercises and fun problems. For this we will use a very popular textbook "Probability and Statistics for Engineers and Scientists" by Ronald E. Walpole et al. (790 pages!) altogether with popular science book like "The Drunkard's Walk: How Randomness Rules Our Lives" by Leonard Mlodinow (just 250 pages).

### K.12.4 Expected learning outcomes

- Probability and sample (probability) spaces
- Discrete and continuous distribution
- Mean and variance
- Central limit theorem, law of large numbers
- Markov processes and chains
- Linear regression and correlation
- Test statistics $\chi^2$ and Students test
- Information and entropy.

### K.12.5 Expected acquired core competences

- Basic (core) concepts, definitions, theoretical results and applied methods & techniques of Probability Theory and Mathematical Statistics

### K.12.6 Textbook

-

### K.12.7 Reference material

-

### K.12.8 Required computer resources

Any electronic spreadsheet (Excel for example) that provides data sorting and graph & chart drawing.

### K.12.9 Evaluation

- In-class participation 1 point for each individual contribution in the lab class but not more than 10 points in total,
- In-class tests up to 15 points (for each test),
- Mid-term exam up to 40 points
- Final examination up to 50 points.

## K.13 Physics I (mechanics)

- **Course name:** Physics I (mechanics)
- **Course number:** XYZ

### K.13.1 Course Characteristics

#### K.13.1.1 Key concepts of the class

- fundamental concepts of physics for calculating problems of mechanics in
  - statics,
  - dynamics.

#### K.13.1.2 What is the purpose of this course?

This course provides the fundamental concepts of physics, in particular focusing on classical mechanics. In general, the aim of this course is:

- to study physical phenomena and laws of physics, the limits of their applicability, application of laws in the most important practical applications; to get acquainted with the basic physical quantities, to know their definition, meaning, methods and units of their measurement; to imagine the fundamental physical experiments and their role in the development of science; to know the purpose and principles of the most important physical devices;
- to acquire skills of work with devices and equipment of modern physical laboratory; skills of use of various methods of physical measurements and processing of experimental data; skills of carrying out physical and mathematical modeling, and also application of methods of the physical and mathematical analysis to the decision of concrete natural science and technical problems;
- to understand the logical connections between the sections of the course of physics, to develop the idea that physics is a universal basis for the technical Sciences, and that those physical phenomena and processes that are still limited in use in technology, in the future may be at the center of innovative achievements of engineering.

#### K.13.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should know ...

- the basic physical phenomena and processes on which the principles of action of objects of professional activity, areas and possibilities of application of physical effects are based;
- fundamental concepts, laws and theories of classical and modern physics, limits of applicability of basic physical models;
- basic physical quantities and constants, their definitions and units of measurement;
- basic physical quantities and constants, their definitions and units of measurement;
- methods of physical research, including methods of modeling physical processes;
- methods for solving physical problems important for technical applications;
- physical bases of measurements, methods of measurement of physical quantities;
- technologies of work with different types of information;

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to ...

- allocate physical content in systems and devices of different physical nature;
- carry out the correct mathematical description of physical phenomena in technological process;
- build and analyze mathematical models of physical phenomena and processes in solving applied problems;
- solve typical problems in the main branches of physics, using methods of mathematical analysis and modeling;
- apply concepts, physical laws and methods of problem solving to perform technical calculations, analysis and solution of practical problems, research in professional activities;

- to use modern physical equipment and devices in solving practical problems, to use the basic techniques of error estimation and experimental data processing;

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

- methods of analysis of physical phenomena in technical devices and systems;
- skills of practical application of the laws of physics, including in the design of products and processes;
- methods of theoretical research of physical phenomena and processes, construction of mathematical and physical models of real systems, solutions of physical problems;
- skills in the use of basic physical devices;
- methods of experimental physical research (planning, staging and processing of experimental data, including the use of standard software packages);
- skills of applying knowledge in the field of physics to study other disciplines.

#### K.13.1.4 Course evaluation

**Table K.31:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 0 |
| Interim performance assessment | 30 | 30 |
| 0 Exams | 50 | 70 |

If necessary, please indicate freely your course's features in terms of students' performance assessment.

#### K.13.1.5 Grades range

If necessary, please indicate freely your course's grading features.

#### K.13.1.6 Resources and reference material

- Fundamentals of Physics (Halliday and Resnick) 10ed, ISBN 978-1-118-23072-5

**Table K.32:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 85-100 |
| B. Good | 75-89 | 70-84 |
| C. Satisfactory | 60-74 | 50-69 |
| D. Poor | 0-59 | 0-49 |

- Arya A. Introduction to Classical Mechanics, Benjamin Cummings

### K.13.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.33:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Kinematics of particles | 63 |
| 2 | Kinetics of particles | 70 |
| 3 | Kinetics of systems of particles | 70 |
| 4 | Statics | 37 |
| 5 | Oscillations | 37 |

#### K.13.2.1 Section 1

**Section title:** Kinematics of particles

**Topics covered in this section:**

- Mathematical review (vectors)
- Measurements and One Dimension Motion (Along a Straight Line)
- Motion in Two and Three Dimensions

### What forms of evaluation were used to test students' performance in this section?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

### Typical questions for ongoing performance evaluation within this section

1. The position of a particle as it moves along a y axis is given by $y = 2sin(\pi t/4)$, with t in seconds and y in centimeters. (a) What is the average velocity of the particle between t = 0 and t = 2.0 s? (b) What is the instantaneous velocity of the particle at t= 0, 1.0, and 2.0 s? (c) What is the average acceleration of the particle between t = 0 and t = 2.0 s? (d) What is the instantaneous acceleration of the particle at t = 0, 1.0, and 2.0 s?
2. A woman walks 250 m in the direction 30 ° east of north, then 175 m directly east. Find (a) the magnitude and (b) the angle of her final displacement from the starting point. (c) Find the distance she walks. (d) Which is greater, that distance or the magnitude of her displacement?
3. Ship A is located 4.0 km north and 2.5 km east of ship B. Ship A has a velocity of 22 km/h toward the south, and ship B has a velocity of 40 km/h in a direction 37° north of east. (a) What is the velocity of A relative to B in unit-vector notation with toward the east? (b) Write an expression (in terms of and ) for the position of A relative to B as a function of t, where t=0 when the ships are in the positions described above. (c) At what time is the separation between the ships least? (d) What is that least separation?
4. A baseball is hit at Fenway Park in Boston at a point 0.762 m above home plate with an initial velocity of 33.53 m/s directed 55.0 ° above the horizontal. The ball is observed to clear the 11.28-m-high wall in left field (known as the "green monster") 5.00 s after it is hit, at a point just inside the left-field foul line pole. Find (a) the horizontal distance down the left-field foul line from home plate to the wall; (b) the vertical distance by which the ball clears the wall; (c) the horizontal and vertical displacements of the ball with respect to home plate 0.500 s before it clears the wall.

**Typical questions for seminar classes (labs) within this section**

1. Most important in an investigation of an airplane crash by the U.S. National Transportation Safety Board is the data stored on the airplane's flight-data recorder, commonly called the "black box" in spite of its orange coloring and reflective tape.The recorder is engineered to withstand a crash with an average deceleration of magnitude $3400g$ during a time interval of 6.50 ms. In such a crash, if the recorder and airplane have zero speed at the end of that time interval, what is their speed at the beginning of the interval?
2. Two vectors are given by $\mathbf{a}=3\mathbf{i}+5\mathbf{j}$ and $\mathbf{b}=2\mathbf{i}+4\mathbf{j}$. Find (a) $\mathbf{a}\times\mathbf{b}$ (b) $\mathbf{a}\cdot\mathbf{b}$ (c) $(\mathbf{a}+\mathbf{b})\cdot\mathbf{b}$ (d) the component of **a** along the direction of **b**.
3. A cannon located at sea level fires a ball with initial speed 82 m/s and initial angle 45°.The ball lands in the water after traveling a horizontal distance 686 m. How much greater would the horizontal distance have been had the cannon been 30 m higher?
4. An elevator without a ceiling is ascending with a constant speed of 10 m/s. A boy on the elevator shoots a ball directly upward, from a height of 2.0 m above the elevator floor, just as the elevator floor is 28 m above the ground.The initial speed of the ball with respect to the elevator is 20 m/s. (a) What maximum height above the ground does the ball reach? (b) How long does the ball take to return to the elevator floor?
5. A football player punts the football so that it will have a "hang time" (time of flight) of 4.5 s and land 46 m away. If the ball leaves the player's foot 150 cm above the ground, what must be the (a) magnitude and (b) angle (relative to the horizontal) of the ball's initial velocity?

**Test questions for final assessment in this section**

1. Two ships are moving parallel to each other in opposite directions with speeds $V_1$ and $V_2$. One ship shoots at the other. Find the angle of a gun to hit the target at the moment when distant between the ships are closest? The speed of the projectile $V_0$ is constant.
2. The velocity vector of a moving body is always parallel to acceleration vector. What is the trajectory of this body?

3. An object moves with non-constant velocity. Can the average velocity over a time interval $(t, t+T)$ be greater than or equal to the maximum instantaneous velocity at this time interval? Prove your answer.
4. A car starts moving with the initial zero velocity and with the acceleration, which depends on the time as $a(t) = 2(1-exp(-t/15))$. Find the average velocity of the car over a time interval 10 s to 40 s.
5. A stone thrown at an angle $\alpha = 30°$ relative to the horizon has the same height H at moments $t_1 = 3$ s and $t_2 = 5$ s after start of his flying. Find the initial stone speed $v_0$ and height $H$.
6. A right angle is drawn on a paper. The ruler being always perpendicular to the bisector of this angle moves along this bisector at a speed of 10 cm/s. The ends of the ruler intersect the sides of the drawn angle. What is the velocity of the intersection points moving along the sides of the right angle relative to the paper?

#### K.13.2.2 Section 2

**Section title:** Kinetics of particles

**Topics covered in this section:**

- Force and Motion
- Kinetic Energy and Work
- Potential Energy and Conservation of Energy

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. A vertical force **F** is applied to a block of mass m that lies on a floor.What happens to the magnitude of the normal force $F_N$ on the block from the floor as

magnitude F is increased from zero if force **F** is (a) downward and (b) upward?

2. A 1400 kg jet engine is fastened to the fuselage of a passenger jet by just three bolts (this is the usual practice). Assume that each bolt supports one-third of the load. (a) Calculate the force on each bolt as the plane waits in line for clearance to take off. (b) During flight, the plane encounters turbulence, which suddenly imparts an upward vertical acceleration of 2.6 $m/s^2$ to the plane. Calculate the force on each bolt now.
3. A person riding a Ferris wheel moves through positions at (1) the top, (2) the bottom, and (3) mid-height. If the wheel rotates at a constant rate, rank these three positions according to (a) the magnitude of the person's centripetal acceleration, (b) the magnitude of the net centripetal force on the person, and (c) the magnitude of the normal force on the person, greatest first.
4. A box is on a ramp that is at angle $\theta$ to the horizontal. As $\theta$ is increased from zero, and before the box slips, do the following increase, decrease, or remain the same: (a) the component of the gravitational force on the box, along the ramp, (b) the magnitude of the static frictional force on the box from the ramp, (c) the component of the gravitational force on the box, perpendicular to the ramp, (d) the magnitude of the normal force on the box from the ramp, and (e) the maximum value $f_{s,max}$ of the static frictional force?
5. In three situations, a single force acts on a moving particle. Here are the velocities (at that instant) and the forces: (1) **v**=-4**i**, **F**=6**i**-20**j** (2) **v**=2**i**-3**j**, **F**=-2**i**+7**j** (3) **v**=-3**i**+1**j**, **F**=2**i**+6**j**. Rank the situations according to the rate at which energy is being transferred, greatest transfer to the particle ranked first, greatest transfer from the particle ranked last. **v**=-3**i**+1**j**, **F**=2**i**+6**j**.
6. What is the spring constant of a spring that stores 25 **J** of elastic potential energy when compressed by 7.5 cm?

**Typical questions for seminar classes (labs) within this section**

1. A shot putter launches a 7.260 kg shot by pushing it along a straight line of length 1.650 m and at an angle of 34.10° from the horizontal, accelerating the shot to the launch speed from its initial speed of 2.500 m/s (which is due to the athlete's preliminary motion).The shot leaves the hand at a height of 2.110 m and at an

angle of 34.10°, and it lands at a horizontal distance of 15.90 m. What is the magnitude of the athlete's average force on the shot during the acceleration phase? (Hint: Treat the motion during the acceleration phase as though it were along a ramp at the given angle.)

2. A 1000 kg boat is traveling at 90 km/h when its engine is shut off. The magnitude of the frictional force $f_k$ between boat and water is proportional to the speed $v$ of the boat: $f_k = 70v$, where $v$ is in meters per second and $f_k$ is in newtons. Find the time required for the boat to slow to 45 km/h.
3. A police officer in hot pursuit drives her car through a circular turn of radius 300 m with a constant speed of 80.0 km/h. Her mass is 55.0 kg. What are (a) the magnitude and (b) the angle (relative to vertical) of the net force of the officer on the car seat? (Hint: Consider both horizontal and vertical forces.)
4. A 0.250 kg block of cheese lies on the floor of a 900 kg elevator cab that is being pulled upward by a cable through distance $d_1 = 2.40$ m and then through distance $d_2 = 10.5$ m. (a) Through d1, if the normal force on the block from the floor has constant magnitude $F_N = 3.00$ N, how much work is done on the cab by the force from the cable? (b) Through $d_2$, if the work done on the cab by the (constant) force from the cable is 92.61 kJ, what is the magnitude of $F_N$?
5. A block attached to a spring lies on a horizontal frictionless surface. The other end of the spring attached to the wall. The spring constant is 50 N/m. Initially, the spring is at its relaxed length and the block is stationary at position $x = 0$. Then an applied force with a constant magnitude of 3.0 N pulls the block in the positive direction of the $x$ axis, stretching the spring until the block stops.When that stopping point is reached, what are (a) the position of the block, (b) the work that has been done on the block by the applied force, and (c) the work that has been done on the block by the spring force? During the block's displacement, what are (d) the block's position when its kinetic energy is maximum and (e) the value of that maximum kinetic energy?
6. A funny car accelerates from rest through a measured track distance in time $T$ with the engine operating at a constant power $P$. If the track crew can increase the engine power by a differential amount $dP$, what is the change in the time required for the run?
7. A spring with $k = 100$ N/m is located at the top of a frictionless incline of angle

$37\,°$. The lower end of the incline is distance $D = 1.00$ m from the end of the spring, which is at its relaxed length. A 2.00 kg canister is pushed against the spring until the spring is compressed 0.200 m and released from rest. (a) What is the speed of the canister at the instant the spring returns to its relaxed length (which is when the canister loses contact with the spring)? (b) What is the speed of the canister when it reaches the lower end of the incline?

**Test questions for final assessment in this section**

1. A slab of mass $m_1 = 40$ kg rests on a frictionless floor, and a block of mass $m_2 =$ 10 kg rests on top of the slab. Between block and slab, the coefficient of static friction is 0.60, and the coefficient of kinetic friction is 0.40. A horizontal force **F** of magnitude 100 N begins to pull directly on the block, as shown. In unit-vector notation, what are the resulting accelerations of (a) the block and (b) the slab?
2. A box of canned goods slides down a ramp from street level into the basement of a grocery store with acceleration 0.75 $m/s^2$ directed down the ramp. The ramp makes an angle of 40° with the horizontal.What is the coefficient of kinetic friction between the box and the ramp?
3. A circular curve of highway is designed for traffic moving at 60 km/h. Assume the traffic consists of cars without negative lift. (a) If the radius of the curve is 150 m, what is the correct angle of banking of the road? (b) If the curve were not banked, what would be the minimum coefficient of friction between tires and road that would keep traffic from skidding out of the turn when traveling at 60 km/h?
4. An initially stationary 2.0 kg object accelerates horizontally and uniformly to a speed of 10 m/s in 3.0 s. (a) In that 3.0 s interval, how much work is done on the object by the force accelerating it? What is the instantaneous power due to that force (b) at the end of the interval and (c) at the end of the first half of the interval?
5. An iceboat is at rest on a frictionless frozen lake when a sudden wind exerts a constant force of 200 N, toward the east, on the boat. Due to the angle of the sail, the wind causes the boat to slide in a straight line for a distance of 8.0 m in a direction 20 ° north of east. What is the kinetic energy of the iceboat at the end

of that 8.0 m?

6. A boy is initially seated on the top of a hemispherical ice mound of radius $R = 13.8$ m. He begins to slide down the ice, with a negligible initial speed. Approximate the ice as being frictionless. At what height does the boy lose contact with the ice?
7. The cable of the 1800 kg elevator cab snaps when the cab is at rest at the first floor, where the cab bottom is a distance $d = 3.7$ m above a spring of spring constant $k = 0.15$ MN/m. A safety device clamps the cab against guide rails so that a constant frictional force of 4.4 kN opposes the cab's motion. (a) Find the speed of the cab just before it hits the spring. (b) Find the maximum distance $x$ that the spring is compressed (the frictional force still acts during this compression). (c) Find the distance that the cab will bounce back up the shaft. (d) Using conservation of energy, find the approximate total distance that the cab will move before coming to rest. (Assume that the frictional force on the cab is negligible when the cab is stationary.)

#### K.13.2.3 Section 3

**Section title:** Kinetics of systems of particles

**Topics covered in this section:**

- Center of Mass and Linear Momentum
- Rotation
- Rolling, Torque, and Angular Momentum

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Ricardo, of mass 80 kg, and Carmelita, who is lighter, are enjoying Lake Merced at dusk in a 30 kg canoe.When the canoe is at rest in the placid water, they exchange seats, which are 3.0 m apart and symmetrically located with respect to the canoe's center. If the canoe moves 40 cm horizontally relative to a pier post, what is Carmelita's mass?
2. A steel ball of mass 0.500 kg is fastened to a cord that is 70.0 cm long and fixed at the far end.The ball is then released when the cord is horizontal. At the bottom of its path, the ball strikes a 2.50 kg steel block initially at rest on a frictionless surface. The collision is elastic. Find (a) the speed of the ball and (b) the speed of the block, both just after the collision.
3. A tall, cylindrical chimney falls over when its base is ruptured. Treat the chimney as a thin rod of length 55.0 m.At the instant it makes an angle of 35.0° with the vertical as it falls, what are (a) the radial acceleration of the top, and (b) the tangential acceleration of the top. (Hint: Use energy considerations, not a torque.) (c) At what angle u is the tangential acceleration equal to g?
4. A cannonball and a marble roll smoothly from rest down an incline. Is the cannonball's (a) time to the bottom and (b) translational kinetic energy at the bottom more than, less than, or the same as the marble's?
5. A solid brass cylinder and a solid wood cylinder have the same radius and mass (the wood cylinder is longer). Released together from rest, they roll down an incline. (a) Which cylinder reaches the bottom first, or do they tie? (b) The wood cylinder is then shortened to match the length of the brass cylinder, and the brass cylinder is drilled out along its long (central) axis to match the mass of the wood cylinder.Which cylinder now wins the race, or do they tie?

**Typical questions for seminar classes (labs) within this section**

1. Particle $A$ and particle $B$ are held together with a compressed spring between them.When they are released, the spring pushes them apart, and they then fly off in opposite directions, free of the spring.The mass of $A$ is 2.00 times the mass of $B$, and the energy stored in the spring was 60 J. Assume that the spring has negligible mass and that all its stored energy is transferred to the particles. Once

that transfer is complete, what are the kinetic energies of (a) particle $A$ and (b) particle $B$?

2. Block 2 (mass 1.0 kg) is at rest on a frictionless surface and touching the end of an unstretched spring of spring constant 200 N/m.The other end of the spring is fixed to a wall. Block 1 (mass 2.0 kg), traveling at speed $v_1 = 4.0$ m/s, collides with block 2, and the two blocks stick together.When the blocks momentarily stop, by what distance is the spring compressed?
3. A 1400 kg car moving at 5.3 m/s is initially traveling north along the positive direction of a $y$ axis. After completing a 90 right-hand turn in 4.6 s, the inattentive operator drives into a tree, which stops the car in 350 ms. In unit-vector notation, what is the impulse on the car (a) due to the turn and (b) due to the collision? What is the magnitude of the average force that acts on the car (c) during the turn and (d) during the collision? (e) What is the direction of the average force during the turn?
4. A pulsar is a rapidly rotating neutron star that emits a radio beam the way a lighthouse emits a light beam.We receive a radio pulse for each rotation of the star.The period T of rotation is found by measuring the time between pulses.The pulsar in the Crab nebula has a period of rotation of $T = 0.033$ s that is increasing at the rate of $1.26 \times 10^{-5} s/y$. (a) What is the pulsar's angular acceleration a? (b) If a is constant, how many years from now will the pulsar stop rotating? (c) The pulsar originated in a supernova explosion seen in the year 1054.Assuming constant a, find the initial $T$.
5. A pulley, with a rotational inertia of $1.0 \times 10^{-3}\ kgm^2$ about its axle and a radius of 10 cm,is acted on by a force applied tangentially at its rim.The force magnitude varies in time as $F = 0.50t + 0.30t^2$, with $F$ in newtons and $t$ in seconds.The pulley is initially at rest.At $t$=3.0 s what are its (a) angular acceleration and (b) angular speed?
6. bowling ball of radius $R = 11$ cm along a lane. The ball slides on the lane with initial speed $v_{com,0} = 8.5$ m/s and initial angular speed $\omega_0 = 0$. The coefficient of kinetic friction between the ball and the lane is 0.21.The kinetic frictional force $f_k$ acting on the ball causes a linear acceleration of the ball while producing a torque that causes an angular acceleration of the ball. When speed $v_com$ has decreased enough and angular speed $\omega$ has increased enough, the ball stops

sliding and then rolls smoothly. (a) What then is $v_com$ in terms of $\omega$? During the sliding, what are the ball's (b) linear acceleration and (c) angular acceleration? (d) How long does the ball slide? (e) How far does the ball slide? (f) What is the linear speed of the ball when smooth rolling begins?

**Test questions for final assessment in this section**

1. A thin solid disc rolls without slipping on the surface of a hemispherical pit. What is depth where the disc pressure on the pit wall is equal to its weight? The radius of the pit $R$ is much larger than the radius of the disc $r$.
2. Firefighters sometimes use a high-pressure fire hose to knock down the door of a burning building. Suppose such a hose delivers 22 kg of water per second at a velocity of 16 m/s. Assuming the water hits and runs straight down to the ground (that is, it doesn't bounce back), what average force is exerted on the door?
3. A rigid hoop of radius $R$ and mass $M$ is lying on a horizontal frictionless table and pivoted at the point $P$. A point-like object of mass m with the velocity $V_i$ collides elastically with the resting hoop. After collision the point-like object moves with an unknown velocity $V_f$ in an opposite direction with respect to its initial motion. Find the linear velocity $V_f$ of the point-like object and the angular velocity $\omega_f$ of the hoop after collision.
4. A homogeneous elastic rod of mass $M$ lies on a smooth horizontal table. An elastic ball of mass m hits the end of the rod, moving at a velocity $v$ perpendicular to the rod. Find the ball velocity at the moment when the deformation energy of the rod and the ball is maximal. The friction between the rod and the table should be neglected; the inertia moment of the rod related to the center of the mass is $I = mL^2/12$, where $L$ is length of the rod.

#### K.13.2.4 Section 4

**Section title:** Statics

**Topics covered in this section:**

- Equilibrium

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. A ladder leans against a frictionless wall but is prevented from falling because of friction between it and the ground. Suppose you shift the base of the ladder toward the wall. Determine whether the following become larger, smaller, or stay the same (in magnitude): (a) the normal force on the ladder from the ground, (b) the force on the ladder from the wall, (c) the static frictional force on the ladder from the ground, and (d) the maximum value $f_{s,max}$ of the static frictional force.
2. An automobile with a mass of 1360 kg has 3.05 m between the front and rear axles. Its center of gravity is located 1.78 m behind the front axle.With the automobile on level ground, determine the magnitude of the force from the ground on (a) each front wheel (assuming equal forces on the front wheels) and (b) each rear wheel (assuming equal forces on the rear wheels).
3. A uniform cubical crate is 0.750 m on each side and weighs 500 N. It rests on a floor with one edge against a very small, fixed obstruction. At what least height above the floor must a horizontal force of magnitude 350 N be applied to the crate to tip it?
4. A trap door in a ceiling is 0.91 m square, has a mass of 11 kg, and is hinged along one side, with a catch at the opposite side. If the center of gravity of the door is 10 cm toward the hinged side from the door's center, what are the magnitudes of the forces exerted by the door on (a) the catch and (b) the hinge?

**Typical questions for seminar classes (labs) within this section**

1. A door has a height of 2.1 m along a y axis that extends vertically upward and a width of 0.91 m along an $x$ axis that extends outward from the hinged edge of the door.A hinge 0.30 m from the top and a hinge 0.30 m from the bottom each

support half the door's mass, which is 27 kg. In unit-vector notation, what are the forces door at (a) the top hinge and (b) the bottom hinge?

2. A cubical box is filled with sand and weighs 890 N.We wish to "roll" the box by pushing horizontally on one of the upper edges. (a) What minimum force is required? (b) What minimum coefficient of static friction between box and floor is required? (c) If there is a more efficient way to roll the box, find the smallest possible force that would have to be applied directly to the box to roll it. (Hint: At the onset of tipping, where is the normal force located?)
3. A crate, in the form of a cube with edge lengths of 1.2 m, contains a piece of machinery; the center of mass of the crate and its contents is located 0.30 m above the crate's geometrical center.The crate rests on a ramp that makes an angle $\theta$ with the horizontal.As $\theta$ is increased from zero, an angle will be reached at which the crate will either tip over or start to slide down the ramp. If the coefficient of static friction $\mu_s = 0.7$ between ramp and crate is 0.60, (a) does the crate tip or slide and (b) at what angle u does this occur? If $\mu_s = 0.7$ , (c) does the crate tip or slide and (d) at what angle $\theta$ does this occur? (Hint: At the onset of tipping, where is the normal force located?)
4. A beam of length L is carried by three men, one man at one end and the other two supporting the beam between them on a crosspiece placed so that the load of the beam is equally divided among the three men. How far from the beam's free end is the crosspiece placed? (Neglect the mass of the crosspiece.)
5. The leaning Tower of Pisa is 59.1 m high and 7.44 m in diameter. The top of the tower is displaced 4.01 m from the vertical. Treat the tower as a uniform, circular cylinder. (a) What additional displacement, measured at the top, would bring the tower to the verge of toppling? (b) What angle would the tower then make with the vertical?

**Test questions for final assessment in this section**

1. A uniform ladder whose length is 5.0 m and whose weight is 400 N leans against a frictionless vertical wall. The coefficient of static friction between the level ground and the foot of the ladder is 0.46.What is the greatest distance the foot of the ladder can be placed from the base of the wall without the ladder immediately slipping?

2. A 73 kg man stands on a level bridge of length $L$. He is at distance $L/4$ from one end. The bridge is uniform and weighs 2.7 kN.What are the magnitudes of the vertical forces on the bridge from its supports at (a) the end farther from him and (b) the nearer end?
3. A uniform cube of side length 8.0 cm rests on a horizontal floor.The coefficient of static friction between cube and floor is $\mu$. A horizontal pull $\mathbf{P}$ is applied perpendicular to one of the vertical faces of the cube, at a distance 7.0 cm above the floor on the vertical midline of the cube face. The magnitude of $\mathbf{P}$ is gradually increased. During that increase, for what values of $\mu$ will the cube eventually (a) begin to slide and (b) begin to tip? (Hint: At the onset of tipping, where is the normal force located?)
4. A pan balance is made up of a rigid, massless rod with a hanging pan attached at each end. The rod is supported at and free to rotate about a point not at its center. It is balanced by unequal masses placed in the two pans.When an unknown mass $m$ is placed in the left pan, it is balanced by a mass $m_1$ placed in the right pan; when the mass $m$ is placed in the right pan, it is balanced by a mass $m_2$ in the left pan. Show that $m = \sqrt{m_1 m_2}$

#### K.13.2.5 Section 5

**Section title:** Oscillations

**Topics covered in this section:**

- Harmonic motion

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the maximum acceleration of a platform that oscillates at amplitude 2.20 cm and frequency 6.60 Hz?
2. Two particles oscillate in simple harmonic motion along a common straight-line segment of length A. Each particle has a period of 1.5 s, but they differ in phase by $\pi$ rad. (a) How far apart are they (in terms of A) 0.50 s after the lagging particle leaves one end of the path? (b) Are they then moving in the same direction, toward each other, or away from each other?
3. A thin uniform rod (mass = 0.50 kg) swings about an axis that passes through one end of the rod and is perpendicular to the plane of the swing. The rod swings with a period of 1.5 s and an angular amplitude of 10°. (a) What is the length of the rod? (b) What is the maximum kinetic energy of the rod as it swings?
4. A 1000 kg car carrying four 82 kg people travels over a "washboard" dirt road with corrugations 4.0 m apart. The car bounces with maximum amplitude when its speed is 16 km/h. When the car stops, and the people get out, by how much does the car body rise on its suspension?

**Typical questions for seminar classes (labs) within this section**

1. A massless spring hangs from the ceiling with a small object attached to its lower end.The object is initially held at rest in a position $y_i$ such that the spring is at its rest length. The object is then released from $y_i$ and oscillates up and down, with its lowest position being 10 cm below $y_i$. (a) What is the frequency of the oscillation? (b) What is the speed of the object when it is 8.0 cm below the initial position? (c) An object of mass 300 g is attached to the first object, after which the system oscillates with half the original frequency. What is the mass of the first object? (d) How far below $y_i$ is the new equilibrium (rest) position with both objects attached to the spring?
2. The suspension system of a 2000 kg automobile "sags" 10 cm when the chassis is placed on it. Also, the oscillation amplitude decreases by 50 % each cycle. Estimate the values of (a) the spring constant k and (b) the damping constant b for the spring and shock absorber system of one wheel, assuming each wheel supports 500 kg.

3. The scale of a spring balance that reads from 0 to 15.0 kg is 12.0 cm long. A package suspended from the balance is found to oscillate vertically with a frequency of 2.00 Hz. (a) What is the spring constant? (b) How much does the package weigh?
4. When a 20 N can is hung from the bottom of a vertical spring, it causes the spring to stretch 20 cm. (a) What is the spring constant? (b) This spring is now placed horizontally on a frictionless table. One end of it is held fixed, and the other end is attached to a 5.0 N can.The can is then moved (stretching the spring) and released from rest.What is the period of the resulting oscillation?

**Test questions for final assessment in this section**

1. pendulum is formed by pivoting a long thin rod about a point on the rod. In a series of experiments, the period is measured as a function of the distance $x$ between the pivot point and the rod's center. (a) If the rod's length is $L = 2.20$ m and its mass is $m = 22.1$ g, what is the minimum period? (b) If $x$ is chosen to minimize the period and then $L$ is increased, does the period increase, decrease, or remain the same? (c) If, instead, $m$ is increased without $L$ increasing, does the period increase, decrease, or remain the same?
2. A block weighing 10.0 N is attached to the lower end of a vertical spring ($k$= 200.0 N/m), the other end of which is attached to a ceiling.The block oscillates vertically and has a kinetic energy of 2.00 J as it passes through the point at which the spring is unstretched. (a) What is the period of the oscillation? (b) Use the law of conservation of energy to determine the maximum distance the block moves both above and below the point at which the spring is unstretched. (These are not necessarily the same.) (c) What is the amplitude of the oscillation? (d) What is the maximum kinetic energy of the block as it oscillates?
3. A simple harmonic oscillator consists of an 0.80 kg block attached to a spring ($k = 200$ N/m). The block slides on a horizontal frictionless surface about the equilibrium point $x = 0$ with a total mechanical energy of 4.0 J. (a) What is the amplitude of the oscillation? (b) How many oscillations does the block complete in 10 s? (c) What is the maximum kinetic energy attained by the block? (d) What is the speed of the block at $x = 0.15$ m?

4. A damped harmonic oscillator consists of a block ($m = 2.00$ kg), a spring ($k = 10.0$ N/m), and a damping force ($F = -bv$). Initially, it oscillates with an amplitude of 25.0 cm; because of the damping, the amplitude falls to three-fourths of this initial value at the completion of four oscillations. (a) What is the value of $b$? (b) How much energy has been "lost" during these four oscillations?

## K.14 Differential Equations

- **Course name:** Differential Equations
- **Course number:** XYZ
- **Subject area:** Math

### K.14.1 Course characteristics

#### K.14.1.1 Key concepts of the class

- Ordinary differential equations
- Basic numerical methods

#### K.14.1.2 What is the purpose of this course?

The course is designed to provide Software Engineers and Computer Scientists by knowledge of basic (core) concepts, definitions, theoretical results and techniques of ordinary differential equations theory, basics of power series and numerical methods, applications of the all above in sciences. All definitions and theorem statements (that will be given in lectures and that are needed to explain the keywords listed above) will be formal, but just few of these theorems will be proven formally. Instead (in the tutorial and practice classes) we will try these definitions and theorems on work with routine exercises and applied problems.

#### K.14.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

- recognize the type of the equation,
- identify the method of analytical solution,
- define an initial value problem,
- list alternative approaches to solving ordinary differential equations,
- match the concrete numerical approach with the necessary level of accuracy.

**- What should a student be able to understand at the end of the course?**

- understand application value of ordinary differential equations,
- explain situation when the analytical solution of an equation cannot be found,
- give the examples of functional series for certain simple functions,
- describe the common goal of the numeric methods,
- restate the given ordinary equation with the Laplace Transform.

**- What should a student be able to apply at the end of the course?**

- solve the given ordinary differential equation analytically (if possible),
- apply the method of the Laplace Transform for the given initial value problem,
- predict the number of terms in series solution of the equation depending on the given accuracy,
- implement a certain numerical method in self-developed computer software.

#### K.14.1.4 Course evaluation

**Table K.34:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 20 |
| Interim performance assessment | 30 | 70 |
| Exams | 50 | 80 |

If necessary, please indicate freely your course's features in terms of students' performance assessment:

**Labs/seminar classes:**

- In-class participation 1 point for each individual contribution in a class but not more than 1 point a week (i.e. 14 points in total for 14 study weeks),
- overall course contribution (to accumulate extra-class activities valuable to the course progress, e.g. a short presentation, book review, very active in-class participation, etc.) up to 6 points.

**Interim performance assessment:**

- in-class tests up to 10 points for each test (i.e. up to 40 points in total for 2 theory and 2 practice tests),
- computational practicum assignment up to 10 points for each task (i.e. up to 30 points for 3 tasks).

**Exams:**

- mid-term exam up to 40 points,
- final examination up to 40 points.

**Overall score:** 170 points (100%).

#### K.14.1.5 Grades range

**Table K.35:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 136-170 |
| B. Good | 75-89 | 102-135 |
| C. Satisfactory | 60-74 | 68-101 |
| D. Poor | 0-59 | 0-68 |

If necessary, please indicate freely your course's grading features:

- A: at least 80% of the overall score;
- B: at least 60% of the overall score;
- C: at least 40% of the overall score;
- D: less than 40% of the overall score.

#### K.14.1.6 Resources and reference material

**Textbook:**

-

**Reference material:**

- 
- 
- 

## K.14.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.36:** Course Sections

| Section Number | Section Title | Lectures (hours) | Seminars (labs) | Self-study | Knowledge evaluation |
|---|---|---|---|---|---|
| 1 | First-order equations and their applications | 12 | 6 | 12 | 4 |
| 2 | Introduction to numeric methods for algebraic and first-order differential equations | 8 | 4 | 22 | 1 |
| 3 | Second-order differential equations and their applications | 8 | 4 | 8 | 2 |
| 4 | Laplace transform | 8 | 4 | 12 | 3 |
| 5 | Series approach to linear differential equations | 8 | 4 | 12 | 0 |
| Final examination | | | | | 2 |

### K.14.2.1 Section 1

**Section title:** First-order equations and their applications

**Topics covered in this section:**

- The simplest type of differential equation
- Separable equation
- Initial value problem
- Homogeneous nonlinear equations, substitutions
- Linear ordinary equations, Bernoulli & Riccati equations
- Examples of applications to modeling the real world problems
- Exact differential equations, integrating factor

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**What forms of evaluation were used to test students' performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

1. What is the type of the first order equation?
2. Is the equation homogeneous or not?
3. Which substitution may be used for solving the given equation?
4. Is the equation linear or not?
5. Which type of the equation have we obtained for the modeled real world problem?
6. Is the equation exact or not?

**Typical questions for seminar classes (labs) within this section**

1. Determine the type of the first order equation and solve it with the use of appropriate method.
2. Find the integrating factor for the given equation.
3. Solve the initial value problem of the first order.
4. Construct a mathematical model of the presented real world problem in terms of differential equations and answer for the specific question about it.

**Test questions for final assessment in this section**

1. Linear first order equation. Integrating factor.
2. Bernoulli & Riccati equations.
3. Homogeneous nonlinear equations equations.
4. Exact equations. Substitutions.

#### K.14.2.2 Section 2

**Section title:** Introduction to numeric methods for algebraic and first-order differential equations

**Topics covered in this section:**

- Method of sections (Newton method)
- Method of tangent lines approximation (Euler method)
- Improved Euler method
- Runge-Kutta methods

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the difference between the methods of sections and tangent line approximations?
2. What is the approximation error for the given method?
3. How to improve the accuracy of Euler method?
4. How to obtain a general formula of the Runge-Kutta methods?

**Typical questions for seminar classes (labs) within this section**

1. For the given initial value problem with the ODE of the first order implement in your favorite programming Euler, improved Euler and general Runge-Kutta methods of solving.
2. Using the developed software construct corresponding approximation of the solution of a given initial value problem (provide the possibility of changing of the initial conditions, implement the exact solution to be able to compare the obtained results).
3. Investigate the convergence of the numerical methods on different grid sizes.
4. Compare approximation errors of these methods plotting the corresponding chart for the dependency of approximation error on a grid size.

**Test questions for final assessment in this section**

1. Newton's approximation method.
2. Euler approximation method.
3. Improved Euler method.

4. Runge-Kutta methods.

#### K.14.2.3 Section 3

**Section title:** Second-order differential equations and their applications

**Topics covered in this section:**

- Homogeneous linear equations.
- Constant coefficient homogeneous equations.
- Constant coefficient non-homogeneous equations.
- A method of undetermined coefficients.
- A method of variation of parameters.
- A method of the reduction of order.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the type of the second order equation?
2. Is the equation homogeneous or not?
3. What is a characteristic equation of differential equation?
4. In which form a general solution may be found?
5. What is the form of the particular solution of non-homogeneous equation?

**Typical questions for seminar classes (labs) within this section**

1. Compose a characteristic equation and find its roots.
2. Find the general of second order equation.

3. Determine the form of a particular solution of the equation and reduce the order.
4. Solve a homogeneous constant coefficient equation.
5. Solve a non-homogeneous constant coefficient equation.

**Test questions for final assessment in this section**

1. Homogeneous linear second order equations.
2. Constant coefficient equations. A method of undetermined coefficients.
3. Constant coefficient equations. A method of variation of parameters.
4. Non-homogeneous linear second order equations. Reduction of order.

### K.14.2.4 Section 4

**Section title:** Laplace transform

**Topics covered in this section:**

- Improper integrals. Convergence / Divergence.
- Laplace transform of a function
- Existence of the Laplace transform.
- Inverse Laplace transform.
- Application of the Laplace transform to solving differential equations.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is an improper integral?

2. How to compose the Laplace transform for a certain function?
3. What is a radius of convergence of the Laplace transform?
4. How to determine the inverse Laplace transform for a given expression?
5. How to apply the method of Laplace transform for solving ordinary differential equations?

**Typical questions for seminar classes (labs) within this section**

1. Find the Laplace transform for a given function. Analyze its radius of convergence.
2. Find the inverse Laplace transform for a given expression.
3. Solve the first order differential equation with the use of a Laplace transform.
4. Solve the second order differential equation with the use of a Laplace transform.

**Test questions for final assessment in this section**

1. Laplace transform, its radius of convergence and properties.
2. Inverse Laplace transform. The method of rational functions.
3. Application of Laplace transform to solving differential equations.

#### K.14.2.5 Section 5

**Section title:** Series approach to linear differential equations

**Topics covered in this section:**

- Functional series.
- Taylor and Maclaurin series.
- Differentiation of power series.
- Series solution of differential equations.

**What forms of evaluation were used to test students' performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

1. What are the power and functional series?
2. How to find the radius of convergence of a series?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

3. What is a Taylor series?
4. How to differentiate a functional series?

### Typical questions for seminar classes (labs) within this section

1. Find the radius of convergence of a given series.
2. Compose the Taylor series for a given function.
3. Solve the first order differential equation with the use of Series approach.
4. Solve the second order differential equation with the use of Series approach.

### Test questions for final assessment in this section

1. Taylor and Maclaurin series as functional series. Radius of convergence.
2. Uniqueness of power series. Its differentiation.
3. Application of power series to solving differential equations

## K.15 Data Modeling and Databases I

- **Course name:** Data Modeling and Databases I
- **Course number:** BS-

### K.15.1 Course characteristics

#### K.15.1.1 Key concepts of the class

- Data Modeling
- Databases

#### K.15.1.2 What is the purpose of this course?

Software development has been paired since early days with management of storage and persistency tools. In this course, the classic approach to relational databases will be presented from the design phase via entity-relationship diagrams to the implementation via queries formulated in Structured Query Language (SQL). Students will learn how to model data and use relational databases.

#### K.15.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define

- Define what software requirements are
- List different commercial and non-commercial DBMS
- Define Entity Relationship model
- List different kind of operators of Relational Algebra
- Define Normal Forms and Functional dependencies
- List basic clauses of SQL queries

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- Describe the basic machinery behind DBMS
- Explain Strengths and weaknesses relational model
- Explain the objectives of the normalization process
- Abstract systems using ER model

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply

- Formally modelling a database system
- Reasoning requirements collected from stakeholders and model them using ER diagrams
- Coding in programming languages and interfacing with SQL APIs
- Using transformation approaches to match ER diagrams and relational model

#### K.15.1.4 Course evaluation

**Table K.37:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 30 |
| Interim performance assessment | 30 | 30 |
| Exams | 50 | 40 |

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

#### K.15.1.5 Grades range

**Table K.38:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 65-79 |
| C. Satisfactory | 60-74 | 50-64 |
| D. Poor | 0-59 | 0-49 |

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table M.73, but it may change slightly (usually reduced) depending on how the semester progresses.

#### K.15.1.6 Resources and reference material

### K.15.2 Textbook

- Handouts supplied by the instructor
- 
- 
- 

### K.15.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.39:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Data Modeling | 48 |
| 2 | Databases | 42 |

#### K.15.3.1 Section 1

**Section title:**

Data Modeling

**Topics covered in this section:**

- Requirements Elicitation and Use Cases
- Entity-Relationship model and diagrams
- Relational Algebra and model
- Mapping ER model into relational model

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Design the ER diagram considering specific requirements
2. Given a ER diagram list all entities, and for each entity indicate whether it is a strong or weak entity
3. Is the connection existing between a weak entity type and its owner called a identifying relationship?
4. Given a query expressed in Relational Algebra explain its functioning and objectives

**Typical questions for seminar classes (labs) within this section**

1. Design the ER diagram considering specific requirements
2. Is it true that an entity has properties which uniquely distinguish that entity?
3. Can an attribute have more than one value?
4. Weak entity sets must have total participation in the identifying relationship set?
5. Are Joins are compound operators involving union, selection, and (sometimes) projection?

**Test questions for final assessment in this section**

1. Design the ER diagram considering specific requirements
2. Given a ER diagram list all entities, and for each entity indicate whether it is a strong or weak entity
3. Should a Software Requirements Document include both user and system requirements?
4. Given a DB schema explaining what specific queries expressed in RA compute

### K.15.3.2 Section 2

**Section title:**

Databases

**Topics covered in this section:**

- SQL
- Hints on NOSQL
- Database Application Development
- Normalization
- Physical Database Organization - hints

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Given a query in RA express it in SQL
2. Given a query in SQL express it in RA
3. Given a schema assess if it is normalized according to some normal form, if not normalize it
4. Provide the definition of the normal forms

**Typical questions for seminar classes (labs) within this section**

1. Given a query in natural language code it in SQL
2. Connecting a programming language to SQL API and execute specific query
3. Given a query in SQL and some specific data determine the output of the query
4. Given a query in RA code it in SQL

**Test questions for final assessment in this section**

1. Given a schema assess if it is normalized according to some normal form, if not normalize it
2. Design the ER diagram considering specific requirements
3. Given a ER diagram transform it into a DB schema
4. Given a query in RA code it in SQL
5. Given a query in SQL and some specific data determine the output of the query

## K.16 Operating Systems

- **Course name:** Operating Systems
- **Course number:** R-01

### K.16.1 Course characteristics

#### K.16.1.1 Key concepts of the class

- Structure of an operating system
- Specific mechanisms, policies, and algorithms used to implement the different parts of an operating system

#### K.16.1.2 What is the purpose of this course?

Operating systems are the core part of a computing device and computing devices are an integral part of our life, not only as programmers, but also just as human being – it is enough to think at smart homes infrastructures, now available at accessible prices to everyone, at car devices, like smart navigator and cruise control systems, at other infrastructures. Therefore, a fundamental understanding of the structure of an operating systems has a paramount role in the curriculum of a student in computer science and engineering. The purpose of this course is to provide such understanding. This is a core course, so it is not among its goals to explore the details of the various proposal for operating systems that are now emerging: this is the subject of more advanced endeavours.

#### K.16.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define:

- fundamental components of an Operating Systems,
- organization of primary memory and the associated concept of virtual memory, with techniques based on paging and segmenting,
- structure of secondary memory (file systems),

- management of the processor(s) and of the connected scheduling algorithms,
- allocation of resources and the associated problems (deadlocks),
- approaches to handle I/O.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples):

- strategies and algorithms for allocating processor(s) time to processes,
- strategies and algorithms for allocating primary memory to processes,
- the fundamental states of a process and how they are reached,
- concept and implementation of the address space of a process, both in the single threaded and in the multi threaded cases.
- techniques to organize files and directories in secondary memory,
- algorithms for a safe concurrent access to resources, preventing or avoiding deadlocks,
- methods for attaching different kind of devices to a computer, also considering different kind of buses.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply:

- strategies for programming at the Operating System level,
- fundamental system calls for process creation, termination,
- fundamental system calls to allocate, change, and deallocate primary memory to processes,
- libraries to handle buffered and unbuffered interconnections with the computer, including files and I/O devices,
- the identification of the most suitable algorithms for process, memory, and I/O management depending on the context in which their target operating system is working.

**Table K.40:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes (weekly evaluations) | 20 | 35 [a] |
| Interim performance assessment | 30 | (merged with the above) |
| Exams | 50 | 65 [b] |

[a]Of which 15 for online test done during the tutorial, 15 for homeworks related to labs and 5 for professor discretion.
[b]Of which 30% for the written and 35% for the oral.

#### K.16.1.4 Course evaluation

Each component apart from homeworks will be assessed on a scale 0-10, where 6 is the minimum passing grade. In case of exceptional work a 10 cum laude will be assigned, with a numeric value from 11 to 13 at discretion of the instructor. The homeworks will be initially graded on a scale 0-2 weekly and then the overall grade will be assembled on a scale 0-10.
The grading, though, is not a simple linear combination of the components above. In particular:

- failing any part of the evaluation will trigger a failure in the entire course,
- the online test done during the tutorial, and the homeworks related to labs will be both averaged on the (n-4) best performances, where the 4 will be used to consider any kind of absence of the student, without requiring any additional documentation; if the student has legitimate reasons to be absent more than 4 times, then supporting document should be provided for all her/his absences,
- if there are not failing components, the final grade will be computed as a weighted average of the components above approximated at the highest second digit and then rounded to the closest integer.

Note that the questions for the exam are taken from the textbook.

#### K.16.1.5 Grades range

The semester starts with the default range as proposed in the Table K.93, but it may change slightly (usually reduced) depending on how the semester progresses. See above.

**Table K.41:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 96-100 |
| B. Good | 75-89 | 66-95 |
| C. Satisfactory | 60-74 | 56-65 |
| D. Poor | 0-59 | 0-55 |

#### K.16.1.6 Resources and reference material

- **Textbook:** Andrew S. Tanenbaum and Herbert Bos. *Modern Operating Systems* (5th Edition), Pearson
- **Reference:** Andrew S. Tanenbaum and David J. Wetherall. Computer Networks (5th Edition), Pearson
- **Reference:** Brian W. Kernighan, Dennis M. Ritchie. The C Programming Language - 2nd Edition, Prentice Hall
- **Reference:** Maurice J.Bach. The design of the Unix Operating System, PRENTICE-HALL, INC., Englewood Cliffs, New Jersey 07632

### K.16.2 Course Sections

The course is organized in 15 weeks with every weeks 2 academics hours of lectures, 2 academic hours of labs, and 2 academic hours of tutorials. The main sections of the course and approximate hour distribution between them is as follows:

**Table K.42:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Revision of programming fundamentals for OS | 12 |
| 2 | Processes and threads | 30 |
| 3 | Memory management | 30 |
| 4 | File system, I/O, and management of resources | 18 |

#### K.16.2.1 Section 1

**Section title:**

Revision of programming fundamentals for OS

**Topics covered in this section:**

- Revision of the structure of a C program
- Overall organization of the computation in C
- Preprocessing
- Simulating function calls in preprocessing
- Analogies between macros and call by name
- Meaning of a variable in C
- Scope and extent of a variable
- Managing data structures with variable length
- Allocation and deallocation of memory
- Pointers and pointer arithmentics
- Pointers to functions
- Usage of pointers to function to simulate virtual functions
- Examples of usage of pointers to function in real life scenarios
- Pointers to functions to perform map and reduce

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
| --- | --- |
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Explain the difference between an include file and a library.
2. Is a parameter of a macro a "real" parameter?
3. Discuss the importance of the conditional compilation.
4. What happens when a function returning a pointer returns the address of a local variable?
5. Detail the meaning of the keyword static and external for supporting information hiding.
6. Describe how the use of virtual functions can make the code more flexible.

**Typical questions for seminar classes (labs) within this section**

1. Given a source .c file including a .h header file, show the results of preprossing in terms of the generated c file.
2. Write a macro and a function in C for the same purpose and discuss pros and cons of both approaches.
3. Show how you can write a generic swap function as a macro.
4. Write the code allocating dynamic memory for a 2 dimensional array and initializing it.
5. Provide an example of how with pointers it is possible in the called function to alter values of variable located in the calling function.
6. Using function pointers, write a sorting function having the sorting rule as a parameter of such function.

**Test questions for final assessment in this section**

1. Discuss the difference in the compiled code when using function and when using macros instead.
2. Provide examples of functions that cannot be transformed into macros, also discussing the motivation for such impossibility.
3. Describe the rules for scope and extent for local variables, static variables (in all cases), and pointers, supplying also code examples of them.
4. Detail the structure of the address space of a process when using a three dimensional array allocated as a local variable of a function and when such array is allocated dynamically, also describe the types of the variables in use and how the compiler checks them.
5. Outline the assembly code for a function calling another function passed as a parameter of it.

#### K.16.2.2 Section 2

**Section title:**

Processes and Threads

**Topics covered in this section:**

- Process models
- Process creation and termination
- Process hierarchies
- Process states
- Implementation of processes
- Threads
- Interprocess communication
- Races
- Critical regions, busy waiting, sleep and wakeup
- Semaphores
- Monitors
- Principles of scheduling
- Categories of scheduling algorithms
- Most common approaches for scheduling in interactive systems

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Outline the typical life of a process from creation to termination
2. Present the different possible models of waiting
3. Define the concept of a semaphore and how it can be implemented
4. Explain the concept of a monitor from a programming standpoint and how it relates to modern programming paradigms.
5. Discuss advantages and disadvantages of the different scheduling algorithms

**Typical questions for seminar classes (labs) within this section**

1. Write a shell script that produces a file of sequential numbers by reading the last number in the file, adding 1 to it, and then appending it to the file. Run one instance of the script in the background and one in the foreground, each accessing the same file. Answer the following questions: (a) How long does it take before a race condition manifests itself? (b) What is the critical region? (c) How you can modify the script to prevent the race
2. Write a producer-consumer problem that uses threads and shares a common buffer. However, do not use semaphores or any other synchronization primitives to guard the shared data structures. Just let each thread access them when it wants to. Use sleep and wakeup to handle the full and empty conditions. See how long it takes for a fatal race condition to occur. For example, you might have the producer print a number once in a while. Do not print more than one number every minute because the I/O could affect the race conditions.
3. Write a program that creates a pipe. Have two strings – one should contain some text, the other one should be empty. Transfer a text from the first string to another one using the pipe you created. Show the result.
4. Write a C program that forks a child process, waits for 10 seconds and then sends a SIGTERM signal to the child. The child process should run an infinite loop and print "I'm alive" every second
5. Write the solution for the produced-consumer problem using monitors.

**Test questions for final assessment in this section**

1. *From the textbook:* Multiple jobs can run in parallel and finish faster than if they had run sequentially. Suppose that two jobs, each needing 20 minutes of CPU time, start simultaneously. How long will the last one take to complete if they run sequentially? How long if they run in parallel? Assume 50% I/O wait.
2. *From the textbook:* The readers and writers problem can be formulated in several ways with regard to which category of processes can be started when. Carefully describe three different variations of the problem, each one favoring (or not favoring) some category of processes. For each variation, specify what happens when a reader or a writer becomes ready to access the database, and what

happens when a process is finished?

3. *From the textbook:* Consider a system in which threads are implemented entirely in user space, with the run-time system getting a clock interrupt once a second. Suppose that a clock interrupt occurs while some thread is executing in the run-time system. What problem might occur? Can you suggest a way to solve it?
4. *From the textbook:* In this problem you are to compare reading a file using a single-threaded file server and a multithreaded server. It takes 12 msec to get a request for work, dispatch it, and do the rest of the necessary processing, assuming that the data needed are in the block cache. If a disk operation is needed, as is the case one-third of the time, an additional 75 msec is required, during which time the thread sleeps. How many requests/sec can the server handle if it is single threaded? If it is multithreaded?
5. *From the textbook:* There are five batch jobs: A through E, they arrive at a computer center at almost the same time. They have estimated running times of 10, 6, 2, 4, and 8 minutes. Their (externally determined) priorities are 3, 5, 2, 1, and 4, respectively, with 5 being the highest priority. Consider the following scheduling algorithms: **(a)** Round robin, **(b)** Priority scheduling, **(c)** First-come, first-served (run in order 10, 6, 2, 4, 8), **(d)** Shortest job first. For each mentioned scheduling algorithms, determine the mean process turnaround time. Ignore process switching overhead. For (a), assume that the system is multi-programmed, and that each job gets its fair share of the CPU. For (b) through (d), assume that only one job at a time runs, until it finishes. All jobs are completely CPU bound.

#### K.16.2.3 Section 3

**Section title:**

Memory management

**Topics covered in this section:**

- Address space
- Memory abstraction
- Based and limit registers

- Swapping
- Virtual memory
- Paging
- Implementation of paging
- Page replacement algorithms
- Page faults
- Segmentation
- Segmentation with paging

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are the base and limit registers and what are the problems related to their usage?
2. Can you have swapping in absence of paging? And paging in absence of swapping?
3. What mechanisms and algorithms are available to handle effectively paging?
4. Details advantages and disadvantages of the different page replacement algorithms.
5. Describe the difference between paging and segmenting.
6. Is it possible to combine segmenting and paging? If so, how?

**Typical questions for seminar classes (labs) within this section**

1. Run 'free -t -h' in a Linux shell or 'vmstat' in a macOS one. Discuss the output.
2. Write a C program that runs for 10 seconds. Every second it should: (a) allocate 10 MB of memory – fill it with zeros, (b) sleep for 1 second. Then, compile and run the program in the background (./ex2 &) and run 'vmstat 1' at the same time.

Observe what happens to the memory. Pay attention to si and so fields. *Hint:* use memset(ptr, value, size) to fill the allocated memory.

3. Write a program that simulates a paging system using the ageing algorithm. The number of page frames is a parameter. The sequence of page references should be read from a file. For a given input file, your program should print Hit/Miss ratio.
4. Try to construct a sequence of references that will result in increased or decreased Hit/Miss ratio.
5. *From the textbook:* A machine has 16-bit virtual addresses. Pages are 8 KB. How many entries are needed for a single-level linear page table? Explain your computations.

**Test questions for final assessment in this section**

1. *From the textbook:* A computer provides each process with 65,536 bytes of address space divided into pages of 4096 bytes each. A particular program has a text size of 32,768 bytes, a data size of 16,386 bytes, and a stack size of 15,870 bytes. Will this program fit in the machine's address space? Suppose that instead of 4096 bytes, the page size were 512 bytes, would it then fit? Each page must contain either text, data, or stack, not a mixture of two or three of them.
2. *From the textbook:* You are given the following data about a virtual memory system: **1.** The TLB can hold 1024 entries and can be accessed in 1 clock cycle (1 nsec). **2.** A page table entry can be found in 100 clock cycles or 100 nsec. **3.**The average page replacement time is 6 msec. If page references are handled by the TLB 99% of the time, and only 0.01% lead to a page fault, what is the effective address-translation time?
3. *From the textbook:* A small computer on a smart card has four page frames. At the first clock tick, the R bits are 0111 (page 0 is 0, the rest are 1). At subsequent clock ticks, the values are 1011, 1010, 1101, 0010, 1010, 1100, and 0001. If the aging algorithm is used with an 8-bit counter, give the values of the four counters after the last tick.
4. *From the textbook:* A computer with a 32-bit address uses a two-level page table. Virtual addresses are split into a 9-bit top-level page table field, an 11-bit second-

level page table field, and an offset. How large are the pages and how many are there in the address space?

5. *From the textbook:* A computer has 32-bit virtual addresses and 4-KB pages. The program and data together fit in the lowest page (0–4095) The stack fits in the highest page. How many entries are needed in the page table if traditional (one-level) paging is used? How many page table entries are needed for two-level paging, with 10 bits in each part?

#### K.16.2.4 Section 4

**Section title:**

File system, I/O, and management of resources

**Topics covered in this section:**

- File system
- Files and files types, attributes, and operations
- Paths
- File system layout
- Shared files
- File system backups
- FIle system performances
- General structure of I/O
- Block devices and character devices
- Device drivers
- Memory mapped I/O and Direct Memory Access
- Interrups
- Programmed I/O
- Deadlocks
- Conditions for deadlocks
- Strategies to dead with dealock

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. What is the overall structure of a file system

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

2. How are files and directories organized on a disk and what are the roles of i-nodes, when they are used
3. What are the key differences between block and character devices.
4. How does DMA speeds up the computations?
5. List the major classes of strategies to handle deadlock.

**Typical questions for seminar classes (labs) within this section**

1. Create tmp directory with two empty files (file1, file2). Then, create one hard link named link1 to file1. Write a program that scans tmp directory, locates all i-nodes with a hard link count of two or more and for each such file it should display together all file names that point to the file.
2. Implement a simulated file system that will be fully contained in a single regular file stored on the disk. This disk file will contain directories, i-nodes, free-block information, file data blocks, etc. Choose appropriate algorithms for maintaining free-block information and for allocating data blocks (contiguous, indexed, linked). Your program will accept system commands from the user to create/delete directories, create/delete/open files, read/write from/to a selected file, and to list directory contents.
3. Create a file ex1.txt with a random string in it. Write a C program (ex1.c) that changes the string in ex1.txt to “This is a nice day” by using mmap(). Hints: **(a)** open the file in O_RDWR mode, **(b)** use stat() or fstat() to get the size of the file.
4. Write a C program (ex2.c) using line buffer. Write your code according to the instructions: **(a)** each of the 5 characters of “Hello” string should be put in separate printf(), **(b)** add a 1 sec sleep after every printf(). The output should be a 5 sec wait and then “Hello” printed instantaneously.
5. The tee command reads its standard input until end-of- file, writing a copy of the

input to standard output and to the files named in its command-line arguments. Implement tee using I/O system calls. By default, tee overwrites any existing file with the given name. Implement the -a command-line option (tee -a file), which causes tee to append text to the end of a file if it already exists.

6. Write a C program for deadlock detection algorithm reading the resources available form a file (input.txt). For testing purposes consider 5 processes and 3 type of resources. However, your program must be able to process as many processes and resource types, as needed (check next slide for input file structure description). The output of your program should either say that no deadlock is detected or print out the numbers of processes that are deadlocked.

**Test questions for final assessment in this section**

1. *From the textbook:* Two computer science students, Carolyn and Elinor, are having a discussion about i-nodes. Carolyn maintains that memories have gotten so large and so cheap that when a file is opened, it is simpler and faster just to fetch a new copy of the i-node into the i-node table, rather than search the entire table to see if it is already there. Elinor disagrees. Who is right? Why?
2. *From the textbook:* A typical printed page of text contains 50 lines of 80 characters each. Imagine that a certain printer can print 6 pages per minute and that the time to write a character to the printer's output register is so short it can be ignored. Does it make sense to run this printer using interrupt-driven I/O if each character printed requires an interrupt that takes 50 $\mu$sec all-in to service?
3. *From the textbook:* Consider a disk that has 10 data blocks starting from block 14 through 23. Let there be 2 files on the disk: f1 and f2. The directory structure lists that the first data blocks of f1 and f2 are respectively 22 and 16. The FAT table is as follows: (14,18), (15,17), (16,23), (17,21), (18,20), (19,15), (20,-1), (21,-1), (22,19), (23,14), where (x,y) indicates that the value stored in table entry x points to data block y. What are the data blocks allotted to f1 and f2?
4. *From the textbook:* Explain how hard links and soft links differ with respective to i-node allocations.
5. *From the textbook:* When a user program makes a system call to read or write a disk file, it provides an indication of which file it wants, a pointer to the data

buffer, and the count. Control is then transferred to the operating system, which calls the appropriate driver. Suppose that the driver starts the disk and terminates until an interrupt occurs. In the case of reading from the disk, obviously the caller will have to be blocked (because there are no data for it). What about the case of writing to the disk? Need the caller be blocked awaiting completion of the disk transfer?

6. *From the textbook:* The banker's algorithm is being run in a system with m resource classes and n processes. In the limit of large m and n, the number of operations that must be performed to check a state for safety is proportional to $\mathrm{m}^{a}\mathrm{n}^{b}$ . What are the values of a and b ?

## K.17 Probability and Statistics

- **Course name:** Probability and statistics
- **Course number:** BS-02

### K.17.1 Course Characteristics

#### K.17.1.1 Key concepts of the class

- Probability space & probability basics
- Random variables and their characteristics
- Limit theorems
- Introduction into mathematical statistics

#### K.17.1.2 What is the purpose of this course?

The main idea of the course is to study mathematical basis of modelling random experiments. The course includes constructing a probability space, a model of a random experiment, and its applications to practice. After that, random variables and their properties are considered. As examples of applying this theoretical background, limit theorems of probability theory are proved (law of large numbers, central limit theorem) and some elements of mathematical statistics are studied.

#### K.17.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to:

- construct a mathematical model of a random experiment (probability space)
- calculate conditional probabilities
- use probability generating functions for discrete random variables
- find confidence intervals for parameters of a normal distribution
- estimate unknown parameters of distributions

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to understand:

- probability function and its properties
- law of total probability and Bayes' theorem
- independence of events and of random variables
- different continuous distributions
- multivariate distributions for discrete and continuous cases
- maximum likelihood estimator method

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

- find expected value, variance and other characteristics of a random variable
- apply limit theorems (law of large numbers and central limit theorem)
- find parameters of a simple linear regression

#### K.17.1.4 Course evaluation

**Table K.43:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Test 1 | ? | 10 |
| Midterm | ? | 25 |
| Test 2 | ? | 10 |
| Participation | ? | 5 |
| Final exam | ? | 50 |

#### K.17.1.5 Grades range

Table K.44: Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 85-100 |
| B. Good | 75-89 | 65-84 |
| C. Satisfactory | 60-74 | 45-64 |
| D. Poor | 0-59 | 0-44 |

#### K.17.1.6 Resources and reference material

- Durrett Rick. (2019) Probability. Theory and Examples,
- Suhov Y, Kelbert M (2005) Probability and Statistics by Example, Cambridge University Press

### K.17.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

Table K.45: Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Basics of Probability | 12 |
| 2 | Univariate Distributions | 24 |
| 3 | Multivariate distributions | 24 |
| 4 | Limit theorems & Introduction into Mathematical Statistics | 30 |
| hline | | |

#### K.17.2.1 Section 1

**Section title:** Basics of Probability

**Topics covered in this section:**

- Probability space. $\sigma$-algebra of events. Axiomatic definition of probability
- Classical model of probability

- Independence of events
- Conditional probability
- Probability of a sum of events (of a product of events)
- Law of total probability. Bayes theorem.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Give an example of three events that are pairwise independent, but not mutually independent.
2. Can disjoint events be independent?
3. A six-sided die is rolled. Construct an algebra of events for this random experiment.
4. Derive the formula for calculating the probability of a sum $P(A+B+C)$.

**Typical questions for seminar classes (labs) within this section**

1. Each game of a match between two equal players can end with a victory of one of them with probability 0.5 independently of the other games. Each victory yields one point, and the match is played until one of the players scores 6 points. Due to technical reasons the match was interrupted when the score was $5:3$ in favour of the first player. What do you think is a fair way to distribute the prize between the players?
2. Seventy numbers are chosen at random from integers $1,2,3,\cdot,100$. What is the probability that the largest number chosen is 98?
3. A hospital specialises in curing three types of diseases: $A$, $B$ and $C$. On average, there are 50% of patients who suffer from disease $A$, 30% of patients with disease $B$, and 20% of patients with disease $C$ (each of the patients has exactly one of

these diseases). The probabilities to fully recover from the diseases are equal to 0.95, 0.9 and 0.85 respectively. A patient who came to the hospital recovered completely. What is the probability that he had disease $B$?

4. A white ball is added into an urn that initially contained $n$ balls. It is known that the probabilities of having $0, 1, 2, \ldots n$ white balls (at the start) in the urn are equal to each other. (a) One ball is taken at random from the urn. What is the probability that the ball is white? (b) The ball taken from the urn has turned out to be white. Find the most probable number of white balls that were in the urn from the start.

**Test questions for final assessment in this section**

1. The probabilities for three students to pass the exam are equal to $\frac{11}{12}$, $\frac{11}{14}$ and $\frac{18}{25}$ respectively. Determine the probability that at least one student passes the exam given that they pass or fail independently of each other.
2. One of 10-digit numbers in which digits go in non-increasing order is chosen at random. Find the probability that exactly 4 different digits are used in this number.
3. Two persons play a game. They take turns in rolling a 10-sided fair die. The first one wins as soon as he rolls 9 or 10, whereas the second one wins as soon as he gets no more than 4. (The game goes on until one of the player's winning conditions is met). Determine the probability for the first player to win the game.
4. Two dice are rolled simultaneously. What is the probability that the sum is even given that it is a multiple of 3?
5. There are 5 white balls and 7 green balls in the first urn; 2 white balls and 10 green balls in the second urn. The third urn, that has initially been empty, is filled with the balls: 4 balls are taken from the first urn, 6 balls are taken from the second urn, and they are placed into the third urn. After that, 2 balls are taken at random from the third urn. It turns out that both these balls are green. Determine the probability that these balls originate from different urns.

#### K.17.2.2 Section 2

**Section title:** Univariate Distributions

**Topics covered in this section:**

- Z-test
- Bernoulli trials and their generalisations
- Discrete random variables and their properties.
- Continuous random variables and their properties

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Bernoulli trials
2. Bernoulli trials. The most probable quantity of successes
3. Bernoulli shift
4. Random variables: general definition. Cumulative distribution function
5. Discrete random variables. Expected value and variance
6. Uniform distribution on a finite set
7. Binomial distribution
8. Indicator random variables
9. Geometric distribution
10. Poisson distribution
11. Probability generating function
12. Continuous random variables. Probability density function. Expected value and variance
13. Uniform distribution on a limited interval
14. Exponential distribution

15. Normal distribution
16. Functional dependence of random variables

**Typical questions for seminar classes (labs) within this section**

1. On average 25% students subscribe to the newsletter. Determine the most probable number of subscribers out of (a) 100 students; (b) 103 students.
2. Two players are playing a match (that consists of several games), each of the games can finish in favour of the younger player with probability 0.6 and in favour of the older player with probability 0.4 The younger player has won exactly five games in the first eight games. What is the probability that he started the match with a defeat?
3. Find the range of variance for random variable $\eta$ if its cumulative distribution function is given by $F_\eta(x) = \begin{cases} 0, & x \le 0, \\ 0.3, & 0 < x \le 2, \\ b, & 2 < x \le 6, \\ 1, & x > 6, \end{cases}$ if $b$ is a parameter that belongs to $(0.3; 1)$.
4. Six people entered the lift at the ground floor of a nine-storied house. Find the expected value for (a) the number of stops where exactly one person gets off the lift; (b) the number of stops where exactly two persons leave the lift.
5. Find the expected value and variance of $a^\xi$ given that $\xi \sim Bin(n, p)$.
6. Random variable $Y$ has a uniform distribution on interval $(a; b)$, and $EY =$ Var $Y = 3$. Find $a$ and $b$.

**Test questions for final assessment in this section**

1. Is it possible for random variable $X$ to have a binomial distribution if (a) $EX = 6$ and $VarX = 3$; (b) $EX = 7$ and $VarX = 4$?
2. Let $Y$ be number of sixes and $Z$ be number of fours one gets when rolling six dice. Find the expected value and variance of $Y + Z$.
3. Let $Z$ be a random variable with geometric distribution. Prove that $P(Z = n + k | z > n) = P(Z = k)$ (*lack of memory property of geometric distribution*).

4. Let us consider a sphere of radius $R$ centered at $O$. Point $M$ is chosen at random inside this circle. Random variable $\xi$ is equal to the length of $OM$. Find the cumulative distribution function, probability density, expected value and variance of $\xi$.
5. Random variable $\theta$ is exponentially distributed with parameter $\lambda$. Calculate the probabilities that $\theta$ belongs to intervals $(0;1),(1;2),\ldots,(n-1;n),\ldots$ and show that these probabilities form a geometric sequence. What is the common ratio of this sequence?
6. It is known that $\xi$ is normally distributed random variable, and $P\{|\xi - E\xi| < 1\} = 0.3$. Find the probability that $|\xi - E\xi| < 2$.

#### K.17.2.3 Section 3

**Section title:** Multivariate Distributions

**Topics covered in this section:**

- Discrete multivariate distributions
- Two variate continuous distributions
- Multivariate continuous distributions
- Multivariate normal distribution

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. The joint distribution of $\xi$ and $\eta$ is provided in the table below.

| $\eta\backslash\xi$ | $-1$ | $0$ | $1$ |
|---|---|---|---|
| $-2$ | $\frac{3}{17}$ | $\frac{4}{17}$ | $\frac{1}{17}$ |
| $2$ | $\frac{1}{17}$ | $\frac{5}{17}$ | $\frac{3}{17}$ |

(a) Find marginal distributions of $\xi$ and $\eta$; (b) find expected value and variance for $\xi$ and $\eta$; (c) determine if $\xi$ and $\eta$ are independent; (d) find correlation coefficient of $\xi$ and $\eta$; (d) find conditional expected values $E(\xi|\eta)$ and $E(\eta|\xi)$.

2. A fair coin is flipped thrice, and for every tails obtained we write a plus; for every heads obtained we write a minus. Random variable $\xi$ is equal to the quantity of tails, and random variable $\eta$ is equal to the quantity of sign changes in the sequence. (a) Find marginal distributions of $\xi$ and $\eta$; (b) find expected value and variance for $\xi$ and $\eta$; (c) determine if $\xi$ and $\eta$ are independent; (d) find correlation coefficient of $\xi$ and $\eta$; (d) find conditional expected values $E(\xi|\eta)$ and $E(\eta|\xi)$.
3. $n$ letters have been written and $n$ envelopes have been inscribed for these letters. An absent minded secretary places the letters into envelopes at random and sends them with the evening post. (a) What is the probability that at least one letter reaches its destination? (b) Find the average quantity of letter that reach their destination.

**Typical questions for seminar classes (labs) within this section**

1. How many times does one have to flip a coin to get the results “heads”, “heads” in succession? Is the result going to change if we replace the sequence with “tails”, “heads”?
2. Forty three equally strong sportsmen take part in a ski race; 18 of them belong to club $A$, 10 to club $B$ and 15 to club $C$. What is the average place for (a) the best participant from club $B$; (b) the worst participant from club $B$?
3. How much rolls does one need on average to get a sequence “6”, “6” when rolling a symmetric six-sided die? And if we change this sequence to “6”, “6”, “6”?
4. $\zeta$ is the quantity of threes and $\eta$ is the quantity of odd digits obtained when rolling a fair die $K$ times. Find correlation coefficient between $\eta$ and $\zeta$.

### K.17.2.4 Test questions for final assessment in the course

1. A fair die is rolled until a four is obtained. Find the expected value of a sum obtained in all the rolls.

2. Find a correlation coefficient between the quantity of sixes and the quantity of fives obtained in $K$ rolls of a fair die.
3. Expected value $\mu$ and covariance matrix $\mathcal{K}$ of random vector $\xi = (\xi_1, \xi_2, \xi_3)^T$ are given. Calculate the expected value and variance of $\eta = \xi_1 - \xi_3$.
4. $N$ letters have been written and envelopes have been inscribed for these letters. An absent-minded secretary puts the letters into envelopes at random and sends them with the evening post. Find $E\xi$ and $Var\xi$, where $\xi$ is the number of letters that reached their destination.

#### K.17.2.5 Section 1

**Section title:** Limit Theorems and Introduction into Mathematical Statistics

**Topics covered in this section:**

- Chebyshev's inequality. Law of large numbers
- Central limit theorem
- Estimating unknown distribution parameters
- Simple linear regression

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Is it possible that for some random variable an equality is reached in Chebyshev's inequality?
2. Give an example of a sequence of random variables that satisfies the law of large numbers, and another sequence that does not satisfy it.
3. What is a maximum likelihood estimator? Calculate maximum likelihood estimators for parameters of a normally distributed sample.

4. Are least square estimators of parameters of a linear regression unbiased? Are they consistent?

**Typical questions for seminar classes (labs) within this section**

1. Let $\xi_n \sim \begin{pmatrix} -\sqrt{n} & 0 & \sqrt{n} \\ 1/2n & 1-1/n & 1/2n \end{pmatrix}$. Does this sequence comply the law of large numbers?
2. Let $\xi_n \sim \begin{pmatrix} -n & 0 & n \\ 2^{-n} & 1-2^{-n+1} & 2^{-n} \end{pmatrix}$. Does this sequence comply the law of large numbers?
3. Find a maximum likelihood estimator for a parameter of a sample with Poisson distribution.
4. Prove that sample variance is a biased estimator of variance for a sample of independent identically distributed random variables. Show that Bessel's correction makes it an unbiased estimator.
5. Provide an example of an unbiased estimator that does not have the least mean square error possible.

**Test questions for final assessment in this section**

1. The probabilities for three students to pass the exam are equal to $\frac{11}{12}$, $\frac{11}{14}$ and $\frac{18}{25}$ respectively. Determine the probability that at least one student passes the exam given that they pass or fail independently of each other.
2. Let us consider independent identically distributed random variables with uniform distribution on $(\theta;\theta+3)$. Find the maximum likelihood estimator of $\theta$. Which one of these estimators is unbiased? Justify your answer.
3. Find the smallest possible value of $P\left(|\xi - E\xi| \le 3\sqrt{Var\xi}\right)$.
4. The probability that a new-born baby is a boy is equal to 0.52. Find the interval which contains the quantity of boys out of 10 000 newborn babies with probability 0.98.
5. Prove that for multivariate normal distribution uncorrelatedness implies independence.
6. Use characteristic functions to show that a sum of independent (and not necessarily identically distributed) random variables also has normal distribution.

## K.18 History

- **Course name:** History
- **Course number:** XYZ
- **Knowledge area:** Humanities

### K.18.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 1st year of BS
- **Semester of instruction:** 2nd semester
- **No. of Credits:** 4 ECTS
- **Total workload on average:** 144 hours overall
- **Class lecture hours:** 2 per week
- **Class tutorial hours:** 0
- **Lab hours:** 0
- **Individual lab hours:** 4 per week
- **Frequency:** weekly throughout the semester
- **Grading mode:** letters: A, B, C, D

### K.18.2 Prerequisites

None

### K.18.3 Course outline

The course is designed to form stable ideas about the history of Russia as an integral part of the World History, about the peculiarities of the historical development of Russia in the context of world historical development. The course should contribute to the understanding of the history of Russia as the development of a multinational community, the establishment of ethnic tolerance and civic maturity.

### K.18.4 Expected learning outcomes

- The main problems of theory and the World History and History of Russia;
- The dates of the most important events;
- Historical concepts, terms, concepts
- The ability to extract information from historical sources, apply it to solve cognitive tasks;
- The ability to use the methods of historical description and explanation;
- The ability to apply appraisal skills to determine and justify their attitude to historical and contemporary events
- To have the skills of independent study of historical sources and analysis of socially significant problems and
- To have the culture of historical thinking and the application of the acquired knowledge for understanding, comprehension;
- To understand the historical nature of socio-political processes and events of our time

### K.18.5 Expected acquired core competences

- Russian history
- World history

### K.18.6 Textbook

- История России с древнейших времен. Том I - Том XXIX / Соловьёв С.М./ Издательство "Лань 2013 - https://e.lanbook.com/book/9856#book_name
- История государства и права России: Учебник для вузов/ Толстая А.И./ Издательство "Юстицинформ 2010. - https://e.lanbook.com/book/10685#book_name
- История развития системы государственной безопасности России / Аверченков В.И., Ерохин В.В., Голембиовская О.М./ Издательство "ФЛИНТА 2011. - https://e.lanbook.com/book/20117#book_name

### K.18.7 Reference material

- http://e.lanbook.com/
- https://moodle.university.innopolis.ru/course/view.php?id=14
- http://www.consultant.ru/

### K.18.8 Required computer resources

No computer resources are required for this course.

### K.18.9 Evaluation

Grading system for the course:

- Seminar (practical) classes in the semester - 30 points.
- Tests performed during the semester - 30 points.
- Final exam - 40 points.

The following scale of compliance was observed for scores of a five-point scale:

- 86 points or more - A;
- from 71 to 85 points - B;
- from 55 to 70 points - C;
- 54 points or less - D.

The following is the meaning of the grades:

- A "Excellent" - the learner has deeply and firmly grasped all the program material, exhaustively, consistently, correctly and logically harmoniously describes it, does not find it difficult to answer with the modification of the task, copes with tasks and practical tasks, correctly justifies the decisions made, knows how to do it yourself summarize and present the material, avoiding mistakes.
- B "Good" - the learner firmly knows the program material, intelligently and essentially expounds it, does not allow significant inaccuracies in answering the question, can correctly apply theoretical provisions and possesses the necessary skills and skills in performing practical tasks.

- C "Satisfactory" - the learner has learned only the basic material, but does not know the individual details, admits inaccuracies, insufficiently correct formulations, breaks the sequence in the presentation of the program material and has difficulty in performing practical assignments.
- D "Unsatisfactory" - the trainee does not know a significant part of the program material, admits significant mistakes, and with great difficulties performs practical tasks and tasks.

## K.19 Data Modeling and Databases 2

- **Course name:** Data Modeling and Databases 2
- **Course number:** XYZ
- **Subject area:** Computer science

### K.19.1 Course characteristics

#### K.19.1.1 Key concepts of the class

- How databases are implemented in practice
- What optimization are performed to ensure high performance of DBMS

#### K.19.1.2 What is the purpose of this course?

While the course Data Modelling and Databases I (DMD I) covered the core concepts behind database design and the relational model, there are further considerations that should be addressed to pursue a career in this field. This course will expand upon what it has been presented in DMD course with focus on both software design, under the form of conceptual and logical DB design, and physical optimization, and will introduce concept such us concurrency and NoSQL databases. More attention will be given to the functioning of Database Management Systems (DBMs), looking at the internal implementation details..

#### K.19.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

- How to design, develop and implement a mid-scale relational database for an application domain using a relational DBMS ,
- How DBMSes support different physical database design, implementation, and optimization issues,
- How datafiles and index files are stored and organized,
- How to use persistence's tools in the context of modern software architectures and the Cloud.

**- What should a student be able to understand at the end of the course?**

- Understand physical database design, implementation, and optimization issues,
- Understand how datafiles and index files are stored and organized,
- Understand how to use persistence's tools in the context of modern software architectures and the Cloud.

**- What should a student be able to apply at the end of the course?**

- Manage SQL and Non-SQL databases.

#### K.19.1.4 Course evaluation

**Table K.46:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 30 | 30 |
| Interim performance assessment | 30 | 30 |
| Exams | 40 | 40 |

If necessary, please indicate freely your course's features in terms of students' performance assessment:

**Labs/seminar classes:**

- In-class participation 1 point for each individual contribution in a class but not more than 1 point a week (i.e. 14 points in total for 14 study weeks),
- overall course contribution (to accumulate extra-class activities valuable to the course progress, e.g. a short presentation, book review, very active in-class participation, etc.) up to 6 points.

**Interim performance assessment:**

- in-class tests up to 10 points for each test (i.e. up to 40 points in total for 2 theory and 2 practice tests),
- computational practicum assignment up to 10 points for each task (i.e. up to 30 points for 3 tasks).

**Exams:**

- mid-term exam up to 30 points,
- final examination up to 40 points.

**Overall score:** 100 points (100%).

#### K.19.1.5 Grades range

**Table K.47:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 85-100 | 85-100 |
| B. Good | 75-84 | 75-84 |
| C. Satisfactory | 60-75 | 60-75 |
| D. Poor | 0-59 | 0-59 |

If necessary, please indicate freely your course's grading features:

- A: more than 85 of the overall score;
- B: at least 85 of the overall score;
- C: at least 75 of the overall score;
- D: less than 60 of the overall score.

#### K.19.1.6 Resources and reference material

**Textbook:**

-

**Reference material:**

-
-

### K.19.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.48:** Course Sections

| Section Number | Section Title | Lectures (hours) | Seminars (labs) | Self-study | Knowledge evaluation |
|---|---|---|---|---|---|
| 1 | Database System Concepts and Storage Architecture | 12 | 6 | 12 | 2 |
| 2 | Query Processing | 8 | 4 | 8 | 1 |
| 3 | Transaction Processing | 8 | 4 | 8 | 1 |
| 4 | Advanced Database Concepts | 12 | 6 | 12 | 2 |
| 5 | Special kinds of Databases | 8 | 4 | 8 | 1 |
| Final examination | | | | | 2 |

#### K.19.2.1 Section 1

**Section title:** Database System Concepts and Storage Architecture

**Topics covered in this section:**

- Database System Concepts and Architecture
- Disk Storage, Basic File Structures, Hashing, and Modern Storage Architecture
- Indexing Structures for Files and Physical Database Design

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What kinds of databases you know?
2. How data files can be organized?
3. How index files can be organized?

**Typical questions for seminar classes (labs) within this section**

1. What are the pluses and minuses of the ordered data file approach?

2. What are the pluses and minuses of bitmap indexes?
3. What are the pluses and minuses of hash based indexes?

**Test questions for final assessment in this section**

1. What is a data model in databases?
2. What is the purpose of the DBMS?
3. What is the basic abstraction in RDBMS?
4. What are examples of different database models?

#### K.19.2.2 Section 2

**Section title:** Query Processing

**Topics covered in this section:**

- Strategies for Query Processing
- Query Optimization

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the difference between query tree and query graph?
2. How many query trees can be built for the same query graph?
3. What are the heuristics applied for query optimizations?

**Typical questions for seminar classes (labs) within this section**

1. What "Cost " represent in EXPLAIN ANALYZE Query ?
2. What is "Transactions" and what is the difference between it and queries?
3. Describe Committ and roll back for a transaction ? and how it works ?

**Test questions for final assessment in this section**

1. Definition of the query tree.
2. Definition of the query graph.
3. Heuristic for the query transformations.

#### K.19.2.3 Section 3

**Section title:** Transaction Processing

**Topics covered in this section:**

- Introduction to Transaction Processing
- Concurrency Control Techniques

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are the differences between interleaved and parallel processing of concurrent transactions
2. What are the issues related to concurrent transaction processing?

**Typical questions for seminar classes (labs) within this section**

1. What is transaction processing?
2. What types of transactions do you know?
3. Which transaction performance optimizations do you know?

**Test questions for final assessment in this section**

1. What are mandatory goals of the system log?
2. Which deadlock prevention protocol(s) is/are based on transaction timestamps
3. What is the meaning of the pin/unpin bit?

#### K.19.2.4 Section 4

**Section title:** Advanced Database Concepts

**Topics covered in this section:**

- Distributed Database Concept
- Data Mining Concept
- Database Recovery Techniques

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How system log checkpoints creation can be scheduled?
2. When cascading rollback may occur?
3. What should be done to restore the database after catastrophic failure?

**Typical questions for seminar classes (labs) within this section**

1. What is Apache Hadoop?
2. What is Hadoop MapReduce ? and what is the core functionality for it?
3. What is the Requirements of applications using MapReduce ?

**Test questions for final assessment in this section**

1. What kind of optimization(s) may increase efficiency of recovery in case of deferred update?
2. What is the right sequence of operations for the data mining process?
3. What are the characteristics of classification process in data mining?

#### K.19.2.5 Section 5

**Section title:** Special kinds of Databases

**Topics covered in this section:**

- Financial Time-series Databases.
- Container and Cloud Databases.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Explain the key differences between the Time Series Databases and the general-purpose SQL and No-SQL databases
2. What are the typical operations (requests) performed by a Time Series DB?
3. Explain the differences between CCD and more traditional No-SQL databases

**Typical questions for seminar classes (labs) within this section**

1. What is a rationale for the advent of Container- and Cloud-native Databases?
2. Explain the performance and latency requirements for Time Series Databases, and how they can be satisfied
3. Explain the links between CCD technologies, the Agile software development process and microservices-based software architectures

**Test questions for final assessment in this section**

1. Which Time Series Database solutions do you know? What are their similarities/differences, advantages/disadvantages?
2. Suppose you are a Chief Data Officer for a financial investment firm. Propose a Time Series Database technology to be used for storing your market data and trades, and describe the database schema
3. A financial trading firm needs to control its operations P&L (profit and loss) in real time. Propose a software architecture to that end. Explain how Time Series Databases can be used in that architecture.

4. Provide examples of modern CCD technologies. Is Amazon S3 one of them? Explain why or why not.
5. Which software architecture problems are addressed/solved by CCDs? Do CCDs introduce any new architectural issues on their own?
6. Suppose that you are designing a scalable database solution for data-centric business. You have a choice of using a traditional SQL solution, a No-SQL solution, or a CCD. Explain your decision criteria.

# K.20 Networks

- **Course name:** Networks
- **Course number:** —

## K.20.1 Course characteristics

### K.20.1.1 Key concepts of the class

- Computer networking
- TCP/IP layered structure
- Addressing
- Routing
- Applications
- Transport and application protocols

### K.20.1.2 What is the purpose of this course?

This is an introductory course in computer networks. During the course, students will learn the fundamental principles and techniques that constitute the basis for the modern computer networks design. All relevant level of application will be considered and explained from Personal Area Networks to the Internet considering the most relevant technologies involved. The hardware and software architectures involved in the multilayer OSI reference model structure will be described as well as the most important communication protocols. The basic conceptual elements of network security will be introduced in the final part of this course.

### K.20.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define

- Layered structure of a computer network
- Topologies in computer networks
- Addressing schemes in networks

- Routing algorithms
- Security at different TCP/IP layers
- Nuts and bolts of applications in networks

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to:

- How a network is formed
- How addressing is done at different layers of the networks
- How subnets are created
- How different routing protocols work and which one is better in which scenario
- How medium is accessed on a network
- How different networks operate
- And Demonstrate the knowledge and discuss the overal functioning of a computer network

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply

- Configure networks
- Configure routing
- Perform TCP and UDP connections
- Create networks and subnets
- Write simple network applications through sockets

#### K.20.1.4 Course evaluation

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

#### K.20.1.5 Grades range

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table M.76, but it may change slightly (usually reduced) depending on how the semester progresses.

**Table K.49:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 10 | 20 |
| Quizes | 5 | 25 |
| Exams (Mid and Final) | 70 | 40 |
| Attendance, Participation, and disrection of instructor | 15 | 15 |

**Table K.50:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

#### K.20.1.6 Resources and reference material

- Lecture material
- COMPUTER NETWORKS Fifth Edition ANDREW S. TANENBAUM Vrije Universiteit Amsterdam, The Netherlands DAVID J. WETHERALL University of Washington Seattle, WA PRENTICE HALL
- Online resources

### K.20.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.51:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Application layer | 15 |
| 2 | Transport layer | 15 |
| 3 | Network layer | 15 |
| 4 | Physical and data link layer | 15 |
| 5 | Network security | 10 |

### K.20.2.1 Section 1

**Section title:**

Application layer

**Topics covered in this section:**

- Application layer functionality
- Application layer protocols
- Domain name system
- Web
- File transfer

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How application layer protocols work?
2. How DNS work with the transport layer protocols?
3. How does http and https work?
4. How does ftp work?
5. How security is incoporated at application layer?
6. Which applications need TCP and UDP at the lower layer?

**Test questions for final assessment in this section**

1. As above

### K.20.2.2 Section 2

**Section title:**

Transport layer

**Topics covered in this section:**

- TCP and UDP
- Transmission control
- Quality of service
- Delay-tolerant networking

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How does TCP 3-way handshake work?
2. How reliability is achieved in TCP?
3. What are the pros and cons of connectionless and connection-oriented protocols in transport layer?
4. How security is provided at transport layer?

**Typical questions for seminar classes (labs) within this section**

1. Implement TCP connection through sockets
2. Develop simple application incorporating TCP and UDP

**Test questions for final assessment in this section**

1. As above

#### K.20.2.3 Section 3

**Section title:**

Network layer

**Topics covered in this section:**

- IP addressing
- Subneting
- Routing
- Inter-networking
- host-to-host networking

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are different classes of IPv4 address?
2. Why do we need IPv6?
3. How different routing protocols work and why do we need different routing algorithms?
4. How quality of service is provided at network layer?
5. How to create different subnets?

**Typical questions for seminar classes (labs) within this section**

1. Configure routing table
2. Create subnets
3. Work on quality of service, congestion and related topics

**Test questions for final assessment in this section**

1. As above

#### K.20.2.4 Section 4

**Section title:**

Physical and data link layer

**Topics covered in this section:**

- Layered structure of network
- Physical layer
- Data link layer
- Medium access control
- Congestion
- Swiching
- Sliding window protocols and other protocols in physical and link layer

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How do different protocols work at physical and link layer?
2. How does a switch and a hub work?
3. How does different medium access control mechanisms work?
4. How to detect and correct errors in transmission?
5. What are different transmission media?

**Typical questions for seminar classes (labs) within this section**

1. Configure switches
2. Detect errors in transmission
3. Play with different protocols, e.g., ARP

**Test questions for final assessment in this section**

1. As above

## K.21 Control Theory

- **Course name:** Control Theory
- **Course number:** [S20]

### K.21.1 Course Characteristics

### K.21.2 What subject area does your course (discipline) belong to?

Sensors and actuators; Robotic control.

#### K.21.2.1 Key concepts of the class

- Introduction to Linear Control, Stability of linear dynamical systems
- Controller design
- Sensing, observers, Adaptive control

#### K.21.2.2 What is the purpose of this course?

Linear Control Theory is both an active tool for modern industrial engineering and a prerequisite for most of the state-of-the-art level control techniques and the corresponding courses. With this in mind, the Linear Control course is both building a foundation for the following development of the student as a learner in the fields of Robotics, Control, Nonlinear Dynamics and others, as well as it is one of the essential practical courses in the engineering curricula.

#### K.21.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to outline:

- methods for control synthesis (linear controller gain tuning)
- methods for controller analysis
- methods for sensory data processing for linear systems

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to understand:

- State-space models
- Eigenvalue analysis for linear systems
- Proportional and PD controllers
- How to stabilize a linear system
- Lyapunov Stability
- How to check if the system is controllable
- Observer design
- Sources of sensor noise
- Filters
- Adaptive Control
- Optimal Control
- Linear Quadratic Regulator

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Turn a system of linear differential equations into a state-space model.
- Design a controller by solving Algebraic Riccati eq.
- Find if a system is stable or not, using eigenvalue analysis.

#### K.21.2.4 Course evaluation

**Table K.52:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 30 |
| Interim performance assessment | 30 | 20 |
| Exams | 50 | 50 |

**Table K.53:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 85-100 |
| B. Good | 75-89 | 70-84 |
| C. Satisfactory | 60-74 | 50-69 |
| D. Poor | 0-59 | 0-49 |

#### K.21.2.5 Grades range

#### K.21.2.6 Resources and reference material

Main textbook:

- Ogata, K., 1994. Solving control engineering problems with MATLAB. Englewood Cliffs, NJ: Prentice-Hall.

Other reference material:

- Williams, R.L. and Lawrence, D.A., 2007. Linear state-space control systems. John Wiley & Sons.
- Ogata, K., 1995. Discrete-time control systems (Vol. 2, pp. 446-480). Englewood Cliffs, NJ: Prentice Hall.

### K.21.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.54:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction to Linear Control, Stability of linear dynamical systems | 6 |
| 2 | Controller design | 6 |
| 3 | Sensing, observers, Adaptive control | 6 |

#### K.21.3.1 Section 1

**Section title:** Introduction to Linear Control, Stability of linear dynamical systems

**Topics covered in this section:**

- Control, introduction. Examples.
- Single input single output (SISO) systems. Block diagrams.
- From linear differential equations to state space models.
- DC motor as a linear system.
- Spring-damper as a linear system.
- The concept of stability of the control system. Proof of stability for a linear system with negative real parts of eigenvalues.
- Multi input multi output (MIMO) systems.
- Linear Time Invariant (LTI) systems and their properties.
- Linear Time Varying (LTV) systems and their properties.
- Transfer function representation.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. What is a linear dynamical system?
2. What is an LTI system?
3. What is an LTV system?
4. Provide examples of LTI systems.
5. What is a MIMO system?

**Typical questions for seminar classes (labs) within this section**

1. Simulate a linear dynamic system as a higher order differential equation or in state-space form (Language is a free choice, Python and Google Colab are recommended. Use built-in solvers or implement Runge-Kutta or Euler method.

**Test questions for final assessment in this section**

1. Convert a linear differential equation into a state space form.
2. Convert a transfer function into a state space form.
3. Convert a linear differential equation into a transfer function.
4. What does it mean for a linear differential equation to be stable?

#### K.21.3.2 Section 2

**Section title:** Controller design.

**Topics covered in this section:**

- Stabilizing control. Control error.
- Proportional control.
- PD control. Order of a system and order of the controller.
- PID control.
- P, PD and PID control for DC motor.
- Trajectory tracking. Control input types. Standard inputs (Heaviside step function, Dirac delta function, sine wave).
- Tuning PD and PID. Pole placement.
- Formal statements about stability. Lyapunov theory.
- Types of stability; Lyapunov stability, asymptotic stability, exponential stability.
- Eigenvalues in stability theory. Reasoning about solution of the autonomous linear system.
- Stability proof for PD control.
- Stability in stabilizing control and trajectory tracking.
- Frequency response. Phase response.
- Optimal control of linear systems. From Hamilton-Jacobi-Bellman to algebraic Riccati equation. LQR.
- Stability of LQR.
- Controllability.

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. What is stability in the sense of Lyapunov?
2. What is stabilizing control?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

3. What is trajectory tracking?
4. Why the control for a state-space system does not include the derivative of the state variable in the feedback law?
5. How can a PD controller for a second-order linear mechanical system can be re-written in the state-space form?
6. Write a closed-loop dynamics for an LTI system with a proportional controller.
7. Give stability conditions for an LTI system with a proportional controller.
8. Provide an example of a LTV system with negative eigenvalues that is not stable.
9. Write algebraic Riccati equation for a standard additive quadratic cost.
10. Derive algebraic Riccati equation for a given additive quadratic cost.
11. Derive differential Riccati equation for a standard additive quadratic cost.
12. What is the meaning of the unknown variable in the Riccati equation? What are its property in case of LTI dynamics.
13. What is a frequency response?
14. What is a phase response?

**Typical questions for seminar classes (labs) within this section**

1. Design control for an LTI system using pole placement.
2. Design control for an LTI system using Riccati (LQR).
3. Simulate an LTI system with LQR controller.

**Test questions for final assessment in this section**

1. You have a linear system:
$$\dot{x} = Ax + Bu$$
and a cost function: a) $J = \int (x^\top Qx + u^\top Iu)dt$ b) $J = \int (x^\top Ix + u^\top Ru)dt$ Write Riccati eq. and find LQR gain analytically.

2. You have a linear system a) $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 10 \\ -3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ b) $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 2 & 40 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ Prove whether or not it is stable.
3. You have a linear system a)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 10 \\ -3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

   b)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 2 & 40 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

   Your controller is: a) $\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 100 & 1 \\ 1 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ b) $\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 7 & 2 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ Prove whether the control system is stable.
4. You have linear dynamics:

   a) $2\ddot{q} + 3\dot{q} - 5q = u$ b) $10\ddot{q} - 7\dot{q} + 10q = u$
   c) $15\ddot{q} + 17\dot{q} + 11q = 2u$ d) $20\ddot{q} - \dot{q} - 2q = -u$

   (a) If $u = 0$, which are stable (a - d)?
   (b) Find $u$ that makes the dynamics stable.
   (c) Write transfer functions for the cases $u = 0$ and $u = -100x$.

5. What is the difference between exponential stability, asymptotic stability and optimality?

#### K.21.3.3 Section 3

**Section title:** Sensing, observers, Adaptive control

**Topics covered in this section:**

- Modelling digital sensors: quantization, discretization, lag.
- Modelling sensor noise. Gaussian noise. Additive models. Multiplicative models. Dynamic sensor models.
- Observability.
- Filters.

- State observers.
- Optimal state observer for linear systems.
- Linearization of nonlinear systems.
- Linearization along trajectory.
- Linearization of Inverted pendulum dynamics.
- Model errors. Differences between random disturbances and unmodeled dynamics/processes.
- Adaptive control.
- Control for sets of linear systems.
- Discretization, discretization error.
- Control for discrete linear systems.
- Stability of discrete linear systems.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. What are the sources of sensor noise?
2. How can we combat the lack of sensory information?
3. When it is possible to combat the lack of sensory information?
4. How can we combat the sensory noise?
5. What is an Observer?
6. What is a filter?
7. How is additive noise different from multiplicative noise?

**Typical questions for seminar classes (labs) within this section**

1. Simulate an LTI system with proportional control and sensor noise.
2. Design an observer for an LTI system with proportional control and lack of sensory information.

**Test questions for final assessment in this section**

1. Write a model of a linear system with additive Gaussian noise.
2. Derive and implement an observer.
3. Derive and implement a filter.

# K.22 Introduction to Artificial Intelligence

- **Course name:** Introduction to Artificial Intelligence
- **Course number:** ?????

## K.22.1 Course Characteristics

### K.22.1.1 Key concepts of the class

- Artificial Intelligence: Introduction to the ethical use of AI and the framework of development of AI systems
- Artificial Intelligence: Evolutionary Algorithms

### K.22.1.2 What is the purpose of this course?

Have you ever wondered about how computers decide on what your credit worthiness is, or how they can play chess as good as a world master, or how world class circuits can be built with a minimal number of crossed wires? Perhaps you have wanted to build a human like robot, or have wanted to explore the stars with automated probes. Artificial Intelligence is the field which examines such problems. The goal is to provide a diverse theoretical overview of historical and current thought in the realm of Artificial Intelligence, Computational Intelligence, Robotics and Machine Learning Techniques.

### K.22.1.3 Course objectives based on Bloom's taxonomy

By the end of the course, the students should be able to

- Understand and apply the PEAS model of problem definition
- Understand and apply the Environment Model
- Understand the role of AI within computer science in a variety of fields and applications
- Gather an appreciation of the history of AI founders
- Solve simple problems using random, guided, and directed, search methods and be able to compare their abilities to solve the problem using a statistical argument
- Apply Evolutionary Algorithms to a number of problems

#### K.22.1.4 Course evaluation

- Assignment 1 (20%)
- Assignment 2 (20%)
- Lab Participation (10%)
- Midterm (25%)
- Final (25%)

#### K.22.1.5 Grades range

**Table K.55:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

If necessary, please indicate freely your course's grading features.

#### K.22.1.6 Resources and reference material

- Russell & Norvig - Artificial Intelligence: A Modern Approach, 3rd Edition
- Ashlock - Evolutionary Computation for Modeling and Optimization

### K.22.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.56:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | History and Philosophy of AI | 16 |
| 2 | Searching as Optimization | 16 |
| 3 | Topics in Evolutionary Algorithms | 16 |

#### K.22.2.1 Section 1

**Section title:** History and Philosophy of AI

**Topics covered in this section:**

- Introduction to the practical applications of AI
- History of Epistemology, particularly on the issue of knowledge creation and intelligence.
- Understanding of the Chinese room and Turing test
- Appreciation for the role of AI in Industries and the the application
- Application of the PEAS model
- Application of the Thinking/Acting Humanly/Rationally
- Appreciation of the Ethical Issues in AI

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. State and apply the PEAS model to a set of problems.
2. State the difference between Plato and Aristotle's conceptions of knowledge - refer to Socrates definition?
3. Are you intelligent? What marks you as such? What is the definition?
4. Are you creative? What marks you as such? What is the definition?

**Typical questions for seminar classes (labs) within this section**

1. Apply the PEAS model as a group to a real world instance.
2. Discuss - Humans in many low skilled tasks are being replaced by automation, what role do practitioners have in protecting its abuse
3. Discuss - Asimov's laws of robotics are used as a science fiction application of ethics in AI, do you think they have a role in the real world.
4. Discuss - how can we prevent bias from entering into systems.
5. Discuss - What does it mean for a Computer to be Creative?

**Test questions for final assessment in this section**

1. Apply the PEAS model to a real world instance.
2. You have a classification problem involving images, from the perspective of Plato and Aristotle on knowledge, which algorithm would they implement, justify your decision.
3. Write a short essay based upon either the prosecution or defense of a trial of an Android who has passed the Turing test who is petitioning the court for human rights. What would be your case for or against using concepts in class such as Turing test, Chinese room, etc.?

#### K.22.2.2 Section 2

**Section title:** Title 2

**Topics covered in this section:**

- Searching Algorithms
- Tree Searches and logic, including basics of PROLOG as a lanuage for answering such problems
- First and Second order logic

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Apply Prolog to an example of a family tree
2. Apply Prolog to an example of a logic problem
3. Compare and Contrast between two different search algorithms shown in class and implement.
4. Given an example data set which search would you use and why?

**Typical questions for seminar classes (labs) within this section**

1. How does Prolog implement a tree?
2. Does this program work - mark out any errors.
3. What is the difference between a red and green cut?
4. What is the result of this Query?
5. What is the time complexity of this algorithm?

**Test questions for final assessment in this section**

1. Here is an example family tree. Given a simple Prolog Query - what would be the result?
2. Here is an example logic problem. Given a Prolog Query - what would be the result.
3. Compare and Contrast two different search algorithms in terms of time and space complexity
4. Given data of type X, what search algorithm should you use and why?

#### K.22.2.3 Section 3

**Section title:** Topics in Evolutionary Algorithms

**Topics covered in this section:**

- Understanding of the four base Evolutionary Algorithms: GA, GP, ES, EP
- Application of one of these four.
- Analysis of the application of these four types to a number of problem instances.
- Application of the appropriate statistical models and scientific method (i.e. Hypothesis testing) to evaluate the EA.

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Define a particular EA by its data structure
2. Implement an EA, write a Report about the implementation with sufficient search of the parameter space justified by statistical evaluations
3. Analysis of two EA types base upon their representation and variation operators and suitability to a problem space.

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

4. Produce a new type of EA based upon the concepts seen in class and speculate as to its effectivenesss on a problem via hypothesis testing.

**Typical questions for seminar classes (labs) within this section**

1. Labs within this section are primarily for supporting assistance with the above objectives, e.g. work periods with TA assistance.

**Test questions for final assessment in this section**

1. Show an analysis on a problem instance as to which EA method you would use, justify your answer based on the representation of the problem data.
2. Implement an EA for a problem, show statistical justification of your result.
3. Produce a report about a new creation of an EA system with sufficient justification with hypothesis tests.

# K.23 Software Project

- **Course name:** Software Project (Introduction to Software Engineering from 2021)
- **Course number:** S20-SP

## K.23.1 Course Characteristics

### K.23.1.1 Key concepts of the class

Scientific foundations for Software Engineering depend on the adequate use of methodologies, techniques, tools, and abstractions for developing software. This course explores fundamental techniques for producing, deploying and analyzing software systems from the Agile philosophy perspective.

### K.23.1.2 What is the purpose of this course?

The main objectives of this course are the following:

- To understand agile models of interaction design, development, and project management and put those concepts in practice in real software development.
- To get acquainted with processes, technologies, and activities involved eliciting, analyzing, validating, specifying, and managing functional and non-functional requirements of software systems.
- To manage risks and decisions making based on the risk management
- To be able design software, and to write code that sticks to good practices of cohesion, coupling, modularity, reusing, and encapsulation.
- To get familiar with standard reliability techniques such as validation (testing) and verification.
- To gain practical experience with refactoring techniques to improve the quality of existing code.

### K.23.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to ...

- recall Agile manifesto and principles
- recognize different Agile frameworks
- define core principles and rituals of Scrum framework
- list different types of product backlog items
- recall main concept of system testing, VCM, software architecture, technical debt

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to ...

- discus pros can cons of different project management tools
- describe criteria for good product backlog
- describe criteria for efficient user stories
- explain how to conduct iteration planing, review and retrospective
- discuss possible strategy of Agile implementation

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

- elicit requirements and work with requirements
- design, implement, verify and maintain efficient systems
- present work in front of stakeholders
- effectively critiquing the work of others and receiving such criticism
- work in a development team and collaborate between different teams

### K.23.1.4 Course evaluation

**Group work includes:**

**Table K.57:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Group work | 20 | 60 |
| Individual work | 30 | 40 |
| Test | 50 | 10 |

- **Sprint Delivery.** This course is organized around developing the program in four 2-weeks iterations (including final delivery). The delivery mostly focuses on the presented works that were done (User stories that were finished during the sprit).
- **Artifacts delivery.** During course, besides code delivery students will create eight artifacts (stakeholder interview report, project management tools analysis report, impact mapping, initial project backlog, a definition of done checklist, user stories acceptance criteria, risk breakdown structure, list of quality attribute scenarios) and should follow and update them.
- **Critique Report Delivery.** Students will have an opportunity to evaluate sprints and artifacts delivery of other teams and provide five critique reports that will emphasize what was done well and what needs to improve.
- **Critique Report Reflection.** Students will have an opportunity to update work that was done based on the critique reports from other team and teacher assistant comments and provide critique report reflection that explains changes.

**Individual work includes:**

- **Peer review report.** At the end of the course, students will sum up the contribution of the work and provide a review of the teammates. The evaluation will be based on the quality of the report (1%) creation and teammates reflection (4%).
- **Participation, presentation delivery.** Students that take participation during lecture and labs classes as well as performing during the presentation will get from 1 to 10% of the grade based on the instructor and lector's judgment.
- **Attendance.** It is mandatory to participate in the lectures, labs, and presentations (even if not presenting). Presence on all classes will give 5% to the final grade
- **Optional.** Final grades may be adjusted up to 2% taking into account factors like

personal students' effort, improvements during course, and extra credit problems. This additional 2% is only based on the instructor's personal judgment, and it's not subject to further clarification.

**Test.**

- After the end of the first part of the course (4 weeks), there will be a test that covers all theoretical and practical material that was presented.

#### K.23.1.5 Grades range

**Table K.58:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 85-100 |
| B. Good | 75-89 | 75-84 |
| C. Satisfactory | 60-74 | 60-74 |
| D. Poor | 0-59 | 0-59 |

#### K.23.1.6 Resources and reference material

- Essential Scrum: a practical guide to the most popular agile process by Kenneth S. Rubin.
- Lean Software Development in Action by Janes, Andrea, Succi, Giancarlo
- Software Architecture in Practice, Third Edition by Rick Kazman, Paul Clements, Len Bass
- Design Patterns: Elements of Reusable Object-Oriented Software 1st Edition by Erich Gamma, Richard Helm, Ralph Johnson
- Pro Git by Ben Straub and Scott Chacon

### K.23.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.59:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction | 20 |
| 2 | Iteration 0 | 12 |
| 3 | Iteration 1 | 12 |
| 4 | Iteration 2 | 12 |
| 5 | Iteration 3 | 12 |
| 6 | Iteration 4 | 12 |
| 7 | Conclusion | 4 |

#### K.23.2.1 Section 1

**Section title:** Introduction

**Topics covered in this section:**

- Course explanation
- Transformation in IT Sphere
- Modern trends and the evolution of development processes
- Building IT teams (roles and responsibilities)
- Philosophy of Agile
- Projects presentation

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical tasks and exercises for ongoing performance evaluation within this section**

Given what was discussed in class regarding how people conceptualize their experiences, design an interview script and conduct the interview with a relevant stakeholder for your class project.

1. First, discuss your class project with a stakeholder. With your target stakeholder in mind, design an interview script with closed- and open-ended questions that cover the professional background or knowledge. The questions should also explore in- depth how that person conducts their business in the problem space.
2. Second, prepare to conduct the interview. Plan for 15 minutes and record the interview using an audio recorder. Take brief notes as you go, recognizing that you have the audio recording to complement your notes.
3. Third, transcribe 1-page of information from your interview that describes the most relevant events to your project. This should include the user's goals and any explanation of how and why they pursue those goals.
4. Fourth, choose 60-second sample from audio file that cover most important information and provide (word-to-word) transcription of it.

**Typical tasks and exercises for seminar classes (labs) within this section**

Learn to analyze and compare different programs/tools that already on the market.

1. Create 4-pages document with comparison for tools Project management tools and mind map tools
2. Create 10 minutes presentation with tools observation.

Conduct the interview according to homework assignment

**Test questions for final assessment in this section**

1. What is the project management triangle?
2. What does mean challenging project?
3. What the structure of waterfall model?
4. What the difference between iterative and incremental model?
5. What the key points in scrum process model?
6. What differences between functional, cross-functional and team overlap structures?
7. What difference between semantic vs episodic memory?
8. What are the basis components of a question

### K.23.2.2 Section 2

**Section title:** Iteration 0

**Topics covered in this section:**

- Impact mapping
- Introduction to Product Backlog,
- Assessing criteria's DEEP and INVEST

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical tasks and exercises for ongoing performance evaluation within this section**

Impact map

1. Create an impact map for your project in one of the mind mapping tools.
2. Write a one- to two-page reflection: based on your experience during this exercise, what were the strengths and weaknesses of impact maps? Did you find impact maps helpful to surface any particular assumptions, or were there issues that were difficult to express using this technique?
3. Create a presentation of your impact map. Be sure that it is easy to follow and readable. Prepare for 10 minutes exhibition plus 3 minutes Q&A session.

Product Backlog and INVEST

1. In one document: Provide an overview of your product backlog, your INVEST criteria evaluation and table with user stories that assessed with INVEST technique
2. Create slides to present your overall backlog, user stories and explain your evaluation of the user stories. Prepare for 10 minutes exhibition plus 3 minutes Q&A session.

**Typical tasks and exercises for seminar classes (labs) within this section**

Impact map

1. Create an impact map for one of the project that was presented, ask questions if have problems with creation
2. Present the results and answer to group mate questions

Product Backlog

1. Create a Product Backlog for project that was chosen for impact map creation, ask questions if have problems with creation
2. Create a table with INVEST criteria
3. Apply INVEST technique for product backlog items
4. Present the results and answer to group mate questions

**Test questions for final assessment in this section**

1. Describe the Impact mapping strategy. Reason how it might help to create product backlog.
2. Discuss the difficulties with using impact map technique.
3. What means appropriate level of detail in the backlog?
4. With given user story provide its evaluation with INVEST grading criteria's.

#### K.23.2.3 Section 3

**Section title:** Iteration 1

**Topics covered in this section:**

- Methods for Product Backlog estimation
- Definition of Done (DoD)
- Acceptance criteria (AC)
- Sprint Planning / Review /Retrospective

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical tasks and exercises for ongoing performance evaluation within this section**

Product Backlog estimation, Acceptance criteria, Definition of Done

1. Update your product backlog with clarification of user stories that will come to the Iteration 2.
2. Provide the estimation for your user stories in story points based on the planning poker technique. Use as the basis your team velocity for Iteration 1.
3. Apply for every user story in the product backlog the acceptance criteria in a format that was presented during the lectures.
4. Create a definition of done checklist for your project.
5. Provide brief justification of statements in your checklist.
6. Create slides to present your acceptance criteria for user stories, and your definition of the done checklist.
7. Consider to present it with your Sprint 1 deliveries. The final performance should be no more than 10 minutes plus 3 minutes Q&A session.

**Typical tasks and exercises for seminar classes (labs) within this section**

Product Backlog estimation, Acceptance criteria

1. Update estimation for US and provide AC for the project that was presented last seminar classes, ask questions if have problems
2. Present the results and answer to group mate questions

Scrum rituals discussion

1. Discuss Scrum rituals: Planning, Review, Retrospective

2. In a group create a list with positive and negative outcomes of each ritual
3. Present your list and discuss how to achieve positive outcomes and avoid negative outcomes.

**Test questions for final assessment in this section**

1. Describe sprint rituals. Reason of necessity to follow rituals.
2. Explain how Definition of Done might help to development team. What problems arises when you do not use such technique?
3. What benefits for customer of providing Acceptance Criteria for every user story? What about developers, testers?
4. Is it reasonable to provide Acceptance Criteria for epics stories?
5. When it better to use T-short size estimation, Story Points estimation and Ideal Hours (days)?

#### K.23.2.4 Section 4

**Section title:** Iteration 2

**Topics covered in this section:**

- Practical implementation in industry
- System Testing
- Version Control Management

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical tasks and exercises for ongoing performance evaluation within this section**

Product delivery

1. Implement the tasks that were chosen in the iteration backlog from the product backlog
2. During implementation focus on the process as well and follow the scrum rituals

Critique report

1. Investigate a project report prepared by another team. Team should go over it very carefully on an individual basis, and then meet as a team to discuss conclusions.
2. The team will prepare a brief (appx. 3-5 pages) report summarizing the strengths, weaknesses and recommended actions for the team whose project report are critiquing. (Provide reasoning for artifacts help you to look at all deliveries as a single piece and refreshing the knowledge. Revise deliveries of Sprint. Conduct a code review and analyze a process of the last Sprint of the critiquing team.
3. One member of the team will deliver a 15-minute summary (presentation) of critique to the class.

Critique report reflection

1. Students receive a critique report from another team that attempts to describe the strengths and weaknesses of your artifacts. Every team should first go over their report very carefully on an individual basis and then meet as a team to discuss your conclusions.
2. Update teams project artifacts (impact map, product backlog, INVEST), project management board and source code to include any changes recommended in the critique report that your team finds acceptable.
3. Prepare a brief (appx. 2-4 pages) report summarizing the changes that were made based on the critique report, discussions in class and comments in the evaluation. If changes were not made, the explanation should be provided.

**Typical tasks and exercises for seminar classes (labs) within this section**

System Testing, Version Control Management

1. Examine and create unit tests, integration's tests, UI tests for your project.
2. Example of working with Git (Setup repository, connect it with students project, create master and developer brunches)

Iteration delivery

1. Prepare the agenda for the meeting in advanced including questions to the stakeholder
2. Record the conversation for future analysis;
3. Analyzed Retrospective of the last Iteration, including answers at questions: What was good during the Sprint? What was bad during the Sprint? How do you decide to improve the next sprint?
4. Prepare project managing board, updated to the next Sprint;
5. Prepare project backlog and sprint backlog.

**Test questions for final assessment in this section**

1. Describe differences between Unit test and integration test.
2. Enumerate different types of the test.
3. Explain the purpose of different types of the test.
4. Draw the git workflow. Reason about benefits and weaknesses of presenting workflow.

#### K.23.2.5 Section 5

**Section title:** Iteration 3

**Topics covered in this section:**

- Intro to Software Architecture
- Technical Debt
- SOLID technique

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical tasks and exercises for ongoing performance evaluation within this section**

Product delivery

1. Implement the tasks that were chosen in the iteration backlog from the product backlog
2. During implementation focus on the process as well and follow the scrum rituals

Critique report

1. Investigate a project report prepared by another team. Team should go over it very carefully on an individual basis, and then meet as a team to discuss conclusions.
2. The team will prepare a brief (appx. 3-5 pages) report summarizing the strengths, weaknesses and recommended actions for the team whose project report are critiquing. (Provide reasoning for artifacts help you to look at all deliveries as a single piece and refreshing the knowledge. Revise deliveries of Sprint. Conduct a code review and analyze a process of the last Sprint of the critiquing team.
3. One member of the team will deliver a 15-minute summary (presentation) of critique to the class.

Critique report reflection

1. Students receive a critique report from another team that attempts to describe the strengths and weaknesses of your artifacts. Every team should first go over their report very carefully on an individual basis and then meet as a team to discuss your conclusions.

2. Update teams project artifacts (impact map, product backlog, INVEST), project management board and source code to include any changes recommended in the critique report that your team finds acceptable.
3. Prepare a brief (appx. 2-4 pages) report summarizing the changes that were made based on the critique report, discussions in class and comments in the evaluation. If changes were not made, the explanation should be provided.

**Typical tasks and exercises for seminar classes (labs) within this section**

Technical Debt, SOLID technique

1. Install Sonar Cube and connect it with your project code.
2. Examine the Sonar Cube reports.
3. Reason about your Technical Debt. Was it done intentionally or unintentionally.
4. Discuss your previous experience and/or situation that might happen. Think when the usage of Technical Debt is justified and when it is not.

Iteration delivery

1. Prepare the agenda for the meeting in advanced including questions to the stakeholder
2. Record the conversation for future analysis;
3. Analyzed Retrospective of the last Iteration, including answers at questions: What was good during the Sprint? What was bad during the Sprint? How do you decide to improve the next sprint?
4. Prepare project managing board, updated to the next Sprint;
5. Prepare project backlog and sprint backlog.

**Test questions for final assessment in this section**

1. Enumerate type of Technical Debt
2. Do we always have to pay technical debt?
3. How fast should we pay technical debt?
4. Reason about the situation when software is rarely modified, should we pay technical debt?
5. What is the relationship between product support and technical debt?

6. Explain what acronym SOLID means. Describe two out of for best SOLID practices.

#### K.23.2.6 Section 6

**Section title:** Iteration 4

**Topics covered in this section:**

- Design Patterns
- DevOps philosophy
- Introduction to CI/CD

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical tasks and exercises for ongoing performance evaluation within this section**

Product delivery

1. Implement the tasks that were chosen in the iteration backlog from the product backlog
2. During implementation focus on the process as well and follow the scrum rituals

Critique report

1. Investigate a project report prepared by another team. Team should go over it very carefully on an individual basis, and then meet as a team to discuss conclusions.
2. The team will prepare a brief (appx. 3-5 pages) report summarizing the strengths, weaknesses and recommended actions for the team whose project report are critiquing. (Provide reasoning for artifacts help you to look at all deliveries as a

single piece and refreshing the knowledge. Revise deliveries of Sprint. Conduct a code review and analyze a process of the last Sprint of the critiquing team.
3. One member of the team will deliver a 15-minute summary (presentation) of critique to the class.

Critique report reflection

1. Students receive a critique report from another team that attempts to describe the strengths and weaknesses of your artifacts. Every team should first go over their report very carefully on an individual basis and then meet as a team to discuss your conclusions.
2. Update teams project artifacts (impact map, product backlog, INVEST), project management board and source code to include any changes recommended in the critique report that your team finds acceptable.
3. Prepare a brief (appx. 2-4 pages) report summarizing the changes that were made based on the critique report, discussions in class and comments in the evaluation. If changes were not made, the explanation should be provided.

**Typical tasks and exercises for seminar classes (labs) within this section**

DevOps philosophy, Introduction to CI/CD

1. Examine of working with GitLab
2. Example of working with Jenkins
3. Setup one of the tool, connect it with students project, create a basic pipeline

Iteration delivery

1. Prepare the agenda for the meeting in advanced including questions to the stakeholder
2. Record the conversation for future analysis;
3. Analyzed Retrospective of the last Iteration, including answers at questions: What was good during the Sprint? What was bad during the Sprint? How do you decide to improve the next sprint?
4. Prepare project managing board, updated to the next Sprint;
5. Prepare project backlog and sprint backlog.

**Test questions for final assessment in this section**

1. What the difference between creational, structural and behavioural patterns.
2. Provide some examples of creational patterns
3. Explain what means toolkit and framework, how the differ from each other. Can we use them together?
4. Draw the simple factory method, explain the benefits of that approaches.

### K.23.2.7 Section 7

**Section title:** Conclusion

**Topics covered in this section:**

- Implementing Agile, typical mistakes
- Shu Ha Ri concept

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical tasks and exercises for ongoing performance evaluation within this section**

Final product delivery

1. Implement the tasks that were chosen in the iteration backlog from the product backlog be sure that you have MVP of the product
2. During implementation focus on the process as well and follow the scrum rituals
3. Provide peer review based on the survey

**Typical tasks and exercises for seminar classes (labs) within this section**

Reflection

1. Provide the feedback based on the structure of the course, artifacts, deliveries
2. Reason what material you think you may use in the future
3. What problems you may address with new knowledge?

**Test questions for final assessment in this section NTBU**

Teammates evaluation

1. Asses the contribution to create/improve artifacts
2. Asses the contribution to prepare presentations
3. Asses the contribution in Sprints (developing software)
4. Asses the contribution to create critique reports
5. Asses the contribution to create critique reports reflections
6. Asses the level of conflicts
7. Asses the difficulties of managing person
8. Highlight the areas of improvement (in general)

## K.24 Information Theory

- **Course name:** Information Theory
- **Course number:** XYZ
- **Knowledge area:** Computer Science and Engineering

### K.24.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 3rd year of BS
- **Semester of instruction:** 1st semester
- **No. of Credits:** 4 ECTS
- **Total workload on average:** 144 hours overall
- **Frontal lecture hours:** 2 per week
- **Frontal tutorial hours:** 2 per week
- **Lab hours:** 2 per week
- **Individual lab hours:** 0
- **Frequency:** weekly throughout the semester
- **Grading mode:** letters: A, B, C, D

### K.24.2 Prerequisites

- Probability and statistics
- Calculus
- Introduction to Programming

### K.24.3 Course outline

This course provides a basic introduction to the Information Theory and its applications to digital systems. In particular, it focuses on the theoretical and practical aspects that are connected to the data compression and transmission highlighting the different problems and approaches to overcome the limitations. Moreover, a short introduction about the mathematical requirements will focuses on the probability and basic statistical concepts that will be used throughout the course.

### K.24.4 Expected learning outcomes

- understand key principles of information theory
- understand data compression
- understand data transmission

### K.24.5 Expected acquired core competences

- Discrete probability
- Random variables
- Bayes' theorem
- Entropy
- Mutual information
- Compression
- Source coding
- Huffman coding
- Channel, noise, capacity
- Block codes
- Channel coding
- Error correction
- Advanced applications

### K.24.6 Textbook

-

### K.24.7 Reference material

- Raymond W. Yeung. A First Course in Information Theory. Springer, 2006. ISBN-10: 1408813068. Available for free at: `http://iest2.ie.cuhk.edu.hk/~whyeung/post/draft7.pdf`
- Robert M. Gray. Entropy and Information Theory. Springer, 2011. ISBN-13: 978-1-4419-7969-8. Available for free at: `https://ee.stanford.edu/~gray/it.pdf`

- Stefan M. Moser. Information Theory - Lecture Notes. Available for free at: `http://moser-isi.ethz.ch/scripts.html`

### K.24.8 Required computer resources

No special needs.

### K.24.9 Evaluation

- Tests (60%)
- Project (40%)

# K.25 Software Architectures

- **Course name:** Software Architectures
- **Course number:** XYZ
- **Subject area:** Software Engineering

## K.25.1 Course characteristics

### K.25.1.1 Key concepts of the class

- Introduction to software architecture
- Steps to follow while working with software architecture
- Key architectural patterns
- How to describe software architecture?
- UML as a language for the software architecture

### K.25.1.2 What is the purpose of this course?

Software developers often follow architectural and design patterns informally without a deep understanding of the subject. This prevents the delivered software to work optimally and often to meet functional and nonfunctional requirements. The course provides a detailed understanding of software architectures and how they should be used by software developers and in the software engineering process.

### K.25.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

- Learn how to design high quality software meeting requirements
- Learn how architectural issues can impact software design
- key architectural patterns and their properties

**- What should a student be able to understand at the end of the course?**

- Understand the principles and foundations of software architectures
- Understanding architecture patterns and their use

- Understand formal notations and specifications with related tools

**- What should a student be able to apply at the end of the course?**

- Understand the task and select the proper architecture of the system
- Use particular design tool to create and update architecture of the system

### K.25.1.4 Course evaluation

**Table K.60:** Course grade breakdown

| Type | Default points | Proposed points |
| --- | --- | --- |
| Labs/seminar classes | 50 | 50 |
| Interim performance assessment | 25 | 25 |
| Exams | 25 | 25 |

If necessary, please indicate freely your course's features in terms of students' performance assessment:

**Labs/seminar classes:**

- In-class participation 1 point for each individual contribution in a class but not more than 1 point a week (i.e. 14 points in total for 14 study weeks),
- overall course contribution (to accumulate extra-class activities valuable to the course progress, e.g. a short presentation, book review, very active in-class participation, etc.) up to 6 points.

**Interim performance assessment:**

- in-class tests up to 10 points for each test (i.e. up to 40 points in total for 2 theory and 2 practice tests),
- computational practicum assignment up to 10 points for each task (i.e. up to 30 points for 3 tasks).

**Exams:**

- mid-term exam up to 25 points,
- final examination up to 25 points.

**Overall score:** 100 points (100%).

#### K.25.1.5 Grades range

Table K.61: Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 85-100 | 85-100 |
| B. Good | 75-84 | 75-84 |
| C. Satisfactory | 60-75 | 60-75 |
| D. Poor | 0-59 | 0-59 |

If necessary, please indicate freely your course's grading features:

- A: more than 85 of the overall score;
- B: at least 85 of the overall score;
- C: at least 75 of the overall score;
- D: less than 60 of the overall score.

#### K.25.1.6 Resources and reference material

**Textbook:**

- 

**Reference material:**

- 
- 

### K.25.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

#### K.25.2.1 Section 1

**Section title:** Introduction to software architecture

**Table K.62:** Course Sections

| Section Number | Section Title | Lectures (hours) | Seminars (labs) | Self-study | Knowledge evaluation |
|---|---|---|---|---|---|
| 1 | Introduction to software architecture | 12 | 6 | 12 | 2 |
| 2 | Steps to follow while working with software architecture | 8 | 4 | 8 | 1 |
| 3 | Key architectural patterns | 8 | 4 | 8 | 1 |
| 4 | How to describe software architecture? | 12 | 6 | 12 | 2 |
| 5 | UML as the language for the software architecture | 8 | 4 | 8 | 1 |
| Final examination | | | | | 2 |

### Topics covered in this section:

- Introducing Software Architecture
- How to Become a Software Architect
- Design Thinking Fundamentals

### What forms of evaluation were used to test students' performance in this section?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

### Typical questions for ongoing performance evaluation within this section

1. What is architecture? What is software architecture?
2. What kind of decisions are to be taken based on software architecture?
3. What is the place of the software architecture in the SDLC?

### Typical questions for seminar classes (labs) within this section

1. What activities are be taken by architects?
2. Identify artefacts to be created after analysis stage
3. Identify artefacts to be created after design stage
4. Identify artefacts to be created after evaluation stage
5. Identify artefacts to be created after evolution stage

**Test questions for final assessment in this section**

1. What is requirements specification?
2. What is design specification?
3. What is the classification of specifications?

#### K.25.2.2 Section 2

**Section title:** Steps to follow while working with software architecture

**Topics covered in this section:**

- Devise a Design Strategy
- Stakeholders
- Requirements and architecture choice

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are the responsibilities of software architects?
2. What are the key principles of design thinking?
3. What are the key elements of design mindset?

**Typical questions for seminar classes (labs) within this section**

1. Transform customer requirements into requirements specification for the particular task
2. Highlight how think-do-check cycle works in practice

**Test questions for final assessment in this section**

1. Definition of quality attributes
2. Definition of artifact and stimulus
3. Definition of tactics

#### K.25.2.3 Section 3

**Section title:** Key architectural patterns

**Topics covered in this section:**

- Layers
- Multi-Tier
- Ports and Adaptors

- Pipe and Filter
- Publish-subscriber
- Shared-data
- Broker
- Peer 2 peer
- MVC
- Client-Server
- Master-Slave

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How architectural patterns are defined?
2. Relations between problem-solution and pattern
3. Why practice is important for patterns?

**Typical questions for seminar classes (labs) within this section**

1. Identify the architectural pattern for the particular project
2. Bud a task architecture following a particular pattern

**Test questions for final assessment in this section**

1. What are the key elements of patterns studied?
2. What are the relations between elements of patterns studied?
3. What are the weaknesses of patterns studied?
4. What are the strengths of patterns studied?

### K.25.2.4 Section 4

**Section title:** How to describe software architecture?

**Topics covered in this section:**

- Architectural structures
- Architectural perspectives
- Architectural properties

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are the key architectural structures?
2. What are the key architectural perspectives?
3. What is the meaning of the architectural view?

**Typical questions for seminar classes (labs) within this section**

1. Describe the static perspective of the particular system
2. Describe the dynamic perspective of the particular system
3. Describe the physical perspective of the particular system

**Test questions for final assessment in this section**

1. What are the differences between static, dynamic and physical perspectives?
2. What is coupling, decoupling and cohesion ?
3. What is redundancy? What approaches to resources are?
4. How trade-off can be found?

#### K.25.2.5 Section 5

**Section title:** UML as the language for the software architecture

**Topics covered in this section:**

- Introduction to UML
- The Use Case Diagram
- The Class Diagram
- Collaboration Diagrams
- Sequence Diagram
- State Diagrams
- Package Diagrams
- Component Diagrams
- Deployment Diagrams

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What was the primary reason for UML creation?
2. What other architectural languages exist?

**Typical questions for seminar classes (labs) within this section**

1. Create The Use Case Diagram for the particular task
2. Create The Class Diagram for the particular task
3. Create Collaboration Diagram for the particular task
4. Create Sequence Diagram for the particular task

5. Create State Diagrams for the particular task
6. Create Package Diagrams for the particular task
7. Create Component Diagrams for the particular task
8. Create Deployment Diagrams for the particular task

**Test questions for final assessment in this section**

1. What are the key properties of the The Use Case Diagram diagram?
2. What are the key properties of the The Class Diagram diagram?
3. What are the key properties of the Collaboration Diagrams diagram?
4. What are the key properties of the Sequence Diagram diagram?
5. What are the key properties of the State Diagrams diagram?
6. What are the key properties of the Package Diagrams diagram?
7. What are the key properties of the Component Diagrams diagram?
8. What are the key properties of the Deployment Diagrams diagram?

# K.26 Introduction to Machine Learning

- **Course name:** Introduction to Machine Learning
- **Course number:** R-01

## K.26.1 Course characteristics

### K.26.1.1 Key concepts of the class

- Machine learning paradigms
- Machine Learning approaches, and algorithms

### K.26.1.2 What is the purpose of this course?

There is a growing business need of individuals skilled in artificial intelligence, data analytics, and machine learning. Therefore, the purpose of this course is to provide students with an intensive treatment of a cross-section of the key elements of machine learning, with an emphasis on implementing them in modern programming environments, and using them to solve real-world data science problems.

### K.26.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define

- Different learning paradigms
- A wide variety of learning approaches and algorithms
- Various learning settings
- Performance metrics
- Popular machine learning software tools

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- Difference between different learning paradigms
- Difference between classification and regression
- Concept of learning theory (bias/variance tradeoffs and large margins etc.)
- Kernel methods
- Regularization
- Ensemble Learning
- Neural or Deep Learning

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply

- Classification approaches to solve supervised learning problems
- Clustering approaches to solve unsupervised learning problems
- Ensemble learning to improve a model's performance
- Regularization to improve a model's generalization
- Deep learning algorithms to solve real-world problems

#### K.26.1.4 Course evaluation

**Table K.63:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 0 |
| Interim performance assessment | 30 | 40 |
| Exams | 50 | 60 |

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

#### K.26.1.5 Grades range

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table **??**, but it may change slightly (usually reduced) depending on how the semester progresses.

**Table K.64:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

#### K.26.1.6 Resources and reference material

- T. Hastie, R. Tibshirani, D. Witten and G. James. *An Introduction to Statistical Learning. Springer 2013.*
- *T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning. Springer 2011.*
- *Tom M Mitchel. Machine Learning, McGraw Hill*
- *Christopher M. Bishop. Pattern Recognition and Machine Learning, Springer*

### K.26.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.65:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Supervised Learning | 24 |
| 2 | Decision Trees and Ensemble Learning | 8 |
| 3 | Unsupervised Learning | 8 |
| 4 | Deep Learning | 12 |

#### K.26.2.1 Section 1

**Section title:**

Supervised Learning

**Topics covered in this section:**

- Introduction to Machine Learning
- Derivatives and Cost Function

- Data Pre-processing
- Linear Regression
- Multiple Linear Regression
- Gradient Descent
- Polynomial Regression
- Bias-varaince Tradeoff
- Difference between classification and regression
- Logistic Regression
- Naive Bayes
- KNN
- Confusion Metrics
- Performance Metrics
- Regularization
- Hyperplane Based Classification
- Perceptron Learning Algorithm
- Max-Margin Classification
- Support Vector Machines
- Slack Variables
- Lagrangian Support Vector Machines
- Kernel Trick

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Is it true that in simple linear regression $R^2$ and the squared correlation between X and Y are identical?
2. What are the two assumptions that the Linear regression model makes about the **Error Terms**?

3. Fit a regression model to a given data problem, and support your choice of the model.
4. In a list of given tasks, choose which are regression and which are classification tasks.
5. In a given graphical model of binary random variables, how many parameters are needed to define the Conditional Probability Distributions for this Bayes Net?
6. Write the mathematical form of the minimization objective of Rosenblatt's perceptron learning algorithm for a two-dimensional case.
7. What is perceptron learning algorithm?
8. Write the mathematical form of its minimization objective for a two-dimensional case.
9. What is a max-margin classifier?
10. Explain the role of slack variable in SVM.

**Typical questions for seminar classes (labs) within this section**

1. How to implement various regression models to solve different regression problems?
2. Describe the difference between different types of regression models, their pros and cons, etc.
3. Implement various classification models to solve different classification problems.
4. Describe the difference between Logistic regression and naive bayes.
5. Implement perceptron learning algorithm, SVMs, and its variants to solve different classification problems.
6. Solve a given optimization problem using the Lagrange multiplier method.

**Test questions for final assessment in this section**

1. What does it mean for the standard least squares coefficient estimates of linear regression to be *scale equivariant*?
2. Given a fitted regression model to a dataset, interpret its coefficients.
3. Explain which regression model would be a better fit to model the relationship

between response and predictor in a given data.

4. If the number of training examples goes to infinity, how will it affect the bias and variance of a classification model?
5. Given a two dimensional classification problem, determine if by using Logistic regression and regularization, a linear boundary can be estimated or not.
6. Explain which classification model would be a better fit to for a given classification problem.
7. Consider the Leave-one-out-CV error of standard two-class SVM. Argue that under a given value of slack variable, a given mathematical statement is either correct or incorrect.
8. How does the choice of slack variable affect the bias-variance tradeoff in SVM?
9. Explain which Kernel would be a better fit to be used in SVM for a given data.

#### K.26.2.2 Section 2

**Section title:**

Decision Trees and Ensemble Methods

**Topics covered in this section:**

- Decision Trees
- Bagging
- Boosting
- Random Forest
- Adaboost

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are pros and cons of decision trees over other classification models?

2. Explain how tree-pruning works.
3. What is the purpose of ensemble learning?
4. What is a bootstrap, and what is its role in Ensemble learning?
5. Explain the role of slack variable in SVM.

**Typical questions for seminar classes (labs) within this section**

1. Implement different variants of decision trees to solve different classification problems.
2. Solve a given classification problem problem using an ensemble classifier.
3. Implement Adaboost for a given problem.

**Test questions for final assessment in this section**

1. When a decision tree is grown to full depth, how does it affect tree's bias and variance, and its response to noisy data?
2. Argue if an ensemble model would be a better choice for a given classification problem or not.
3. Given a particular iteration of boosting and other important information, calculate the weights of the Adaboost classifier.

#### K.26.2.3 Section 3

**Section title:**

Unsupervised Learning

**Topics covered in this section:**

- K-means Clustering
- K-means++
- Hierarchical Clustering
- DBSCAN
- Mean-shift

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Which implicit or explicit objective function does K-means implement?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

2. Explain the difference between k-means and k-means++.
3. Whaat is single-linkage and what are its pros and cons?
4. Explain how DBSCAN works.

**Typical questions for seminar classes (labs) within this section**

1. Implement different clustering algorithms to solve to solve different clustering problems.
2. Implement Mean-shift for video tracking

**Test questions for final assessment in this section**

1. K-Means does not explicitly use a fitness function. What are the characteristics of the solutions that K-Means finds? Which fitness function does it implicitly minimize?
2. Suppose we clustered a set of N data points using two different specified clustering algorithms. In both cases we obtained 5 clusters and in both cases the centers of the clusters are exactly the same. Can 3 points that are assigned to different clusters in one method be assigned to the same cluster in the other method?
3. What are the characterics of noise points in DBSCAN?

#### K.26.2.4 Section 4

**Section title:**

Deep Learning

**Topics covered in this section:**

- Artificial Neural Networks

- Back-propagation
- Convolutional Neural Networks
- Autoencoder
- Variatonal Autoencoder
- Generative Adversairal Networks

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is a fully connected feed-forward ANN?
2. Explain different hyperparameters of CNNs.
3. Calculate KL-divergence between two probability distributions.
4. What is a generative model and how is it different from a discriminative model?

**Typical questions for seminar classes (labs) within this section**

1. Implement different types of ANNs to solve to solve different classification problems.
2. Calculate KL-divergence between two probability distributions.
3. Implement different generative models for different problems.

**Test questions for final assessment in this section**

1. Explain what is ReLU, what are its different variants, and what are their pros and cons?
2. Calculate the number of parameters to be learned during training in a CNN, given all important information.
3. Explain how a VAE can be used as a generative model.

# K.27 Distributed Systems and Cloud Computing

- **Course name:** Distributed Systems and Cloud Computing
- **Course number:** X

## K.27.1 Course Characteristics

### K.27.1.1 Key concepts of the class

- Principles of designing and implementing distributed systems
- Structure of distributed system, and algorithms used to implement distributed system

### K.27.1.2 What is the purpose of this course?

This is an introductory course in distributed systems. During the course, students will learn the fundamental principles and techniques that can be applied to design and develop distributed systems, and will cover architectures of distributed systems, communication, naming, fault tolerance, consistency replication, virtualization and security. In addition, we will discuss different aspects of design and implementation of popular distributed systems (such as bittorrent, google file system, HDFS, etc.), programming models (MapReduce, MapReduce2/YARN) and consensus algorithms (Raft and Paxos). The course will not only cover computer science related topics, but will also include a substantial part of software engineering activities.

### K.27.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to understand distributed systems

- Key principles involved in designing and implementing distributed systems
- Distributed Architectures and their management
- Canonical problems and solutions for distributed systems

- Resource sharing, replication and consistency algorithms in distributed environments
- Virtualization, Orchestration and cloud management

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to understand the key components of distributed systems

- Role of different components in distributed systems
- Role of virtualization in distributed systems
- Management and Orchestration of resources in distributed systems
- Fault tolerant approaches in distributed systems

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to develop and implement different components in a distributed environment

- Implementation of access and location transparency in distributed systems
- Designing and implementing consensus algorithms for different distributed environments
- Management of concurrent tasks in distributed settings
- Designing fault tolerant distributed systems

#### K.27.1.4 Course evaluation

**Table K.66:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 20 |
| Interim performance assessment | 30 | 25 |
| Exams | 50 | 55 |

**Table K.67:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 90-100 |
| B. Good | 75-89 | 75-89 |
| C. Satisfactory | 60-74 | 60-74 |
| D. Poor | 0-59 | 0-59 |

#### K.27.1.5 Grades range

#### K.27.1.6 Resources and reference material

-

### K.27.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.68:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction to distributed systems | 16 |
| 2 | Virtualization and cloud computing | 16 |
| 3 | Canonical Problems and Solutions | 24 |

#### K.27.2.1 Section 1

**Section title:** Introduction to distributed systems

**Topics covered in this section:**

- Distributed architectures
- Types of distributed systems
- Processes & Threads
- Multiprocessor and distributed scheduling
- Communication in distributed systems
- Naming in distributed systems

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. State advantages of object based architectures over layer based architectures in distributed system?
2. State advantages of the random walk approach over the flooding approach in unstructured P2P networks for locating the data?
3. Why global variables are not allowed in a RPC?
4. Describe approaches for a server to handle incoming socket connections (at least two).
5. What are pros and cons of different approaches of sockets? When would you choose which one?

**Typical questions for seminar classes (labs) within this section**

1. Find the process ID (PID) of the running program and list of threads associated to that process?
2. Implementation of new threads
3. Performance analysis of given task
4. Implementation of Process and Global Interpreter Lock (GIL)

**Test questions for final assessment in this section**

1. Explain difference between threads and process
2. Define Global Interpreter Lock (GIL) and its role
3. What are remote method invocation (RMI). Please state the benefits and challenges of RMI.
4. Define the role of Marshaling and Unmarshaling in RPC
5. State at least two benefits of both Iterative and recursive naming resolution schemes.

#### K.27.2.2 Section 2

**Section title:** Virtualization and cloud computing

**Topics covered in this section:**

- Foundations of virtualization
- OS-level virtualization
- System level virtualization
- Memory virtualization
- Cloud and data centres

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are containers? State at least three benefits of using them?
2. Why memory reclamation approaches are needed for virtual machines?
3. What are "cgroup" and "namespace" subsystems. How can we use the "cgroup" subsystem?
4. Briefly explain about the "Ballooning" approach and its working to reclaim VM memory?
5. What is a Unikernel? State at least two benefits and one drawback.

**Typical questions for seminar classes (labs) within this section**

1. Hands on Amazon Elastic Compute Cloud (EC2)
2. Set up a public address in Elastic IPs tab and assign it to the web server on which WordPress is hosted.
3. Configure WebServer so that your blog page is accessible only through the public IP address (without specifying the /wordpress path). For example, you specify a public IP address in the address bar of the browser and gain access to the blog.

4. There may be problems with CSS. Search the web for how to fix broken CSS after changing the site URL.

**Test questions for final assessment in this section**

1. State two benefits of immersion cooling over traditional air cooling in data centers?
2. State one main difference of a container from a System Virtual machine?
3. Briefly explain about Copy-on-write storage.
4. What are the roles of “ISA” and “ABI” in operating systems?
5. What is the biggest drawback of Google File system over CephFS?

#### K.27.2.3 Section 3

**Section title:** Canonical Problems and Solutions in DS

**Topics covered in this section:**

- Mutual exclusion
- Leader election
- Clock synchronization
- Consistency issues
- Caching and replication
- Fault Tolerance

**What forms of evaluation were used to test students’ performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Why is Anti-Entropy Protocol considered better than the Gossiping protocol for achieving consistency.

2. What is the role of Heartbeats in RAFT?
3. Explain PAXOS to achieve consistency?
4. What is the benefit of three phase commit over two phase commit? Please specify in terms of coordinator failure?
5. How many replicas are required to identify the fault in Byzantine failure scenarios?

**Typical questions for seminar classes (labs) within this section**

1. In the replication set of databases, what problem may appear if we have an even number of nodes?
2. ICRUD operations in MongoDB.
3. Create simple chat web-application which uses replica set.
4. Explain the steps to shutdown all VPS instances.

**Test questions for final assessment in this section**

1. What is a schema-less data model
2. Why do we need NoSQL? What are its benefits over SQL in terms of ACID and BASE properties?
3. Why is version control recommended? Name at least two version control systems?
4. Role of recovery line and check pointing in distributed snapshot?
5. Explain the difference between Lamport and vector clocks?
6. How the use of buffers at receivers enhance the QoS in networks for streaming media applications? Briefly explain using an example.

## K.28 Compiler Construction

- **Course name:** Compiler Construction
- **Course number:** XYZ
- **Knowledge area:** Software Engineering

### K.28.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 3rd year of BS
- **Semester of instruction:** 1st semester
- **No. of Credits:** 4 ECTS
- **Total workload on average:** 144 hours overall
- **Class lecture hours:** 2 per week
- **Class tutorial hours:** 0 per week
- **Lab hours:** 2 per week
- **Individual lab hours:** 0 per week
- **Frequency:** weekly throughout the semester
- **Grading mode:** letters: A, B, C, D

### K.28.2 Prerequisites

- Introduction to Programming I
- Introduction to Programming II
- Discrete math and logic
- Theoretical Computer Science
- Computer Architecture

### K.28.3 Course outline

The software development process and the depth of programming cannot be understood without a detailed analysis of the compilation process, from the lexical analysis to the syntactical and semantic analysis up to code generation and optimization, and without understanding both strength and limitation of this process. This course dig deeper into this topic building on the fundamental notions studied in theoretical computer science of which is the natural continuation. The typical compiler pipeline will be studied and a project will allow students to practice with the relevant tools.

### K.28.4 Expected learning outcomes

- Understanding in depth the compilation process
- Realizing the limits of the process and of Semantic Analysis
- Read and write grammars for programming language constructs
- Perform lexical analysis and use lexical analyzer generators
- Perform top-down parsing, bottom-up parsing and use parser generators
- Perform semantic analysis

### K.28.5 Expected acquired core competences

- Programming languages and grammars
- BNF and its use
- Automata and their use
- Parsers in the compilation process
- Tools: Flex and Bison
- Static analysis

### K.28.6 Detailed topics covered in the course

The topics below are presented with the granularity of at most the academic hour of instruction. For each topic it is specified if it an **I**ntroduction to the topic, a **D**eep explanation, or a **R**eview of a subject already covered in another course.

- Overview of Compilers
- Lexical Analysis
- Grammars and Parsing
- Bottom-Up Parsing
- Semantic Analysis
- Limitation of semantic analysis

- Symbol Tables & Run-time Systems
- Code Generation & Control Flow Analysis
- Data Flow Analysis
- Optimization

### K.28.7 Textbook

- 
- 

### K.28.8 Reference material

- Slides will be provided during the course

### K.28.9 Required computer resources

Students are required to have notebooks.

### K.28.10 Evaluation

- Mid-term exam (30%)
- Final exam (40%)
- Project (30%)

# K.29 Philosophy II – Language & Perception

- **Course name:** Language & Perception
- **Course number:** ???

## K.29.1 Course characteristics

### K.29.1.1 Key concepts of the class

- Formal philosophy as a systematic bridge among language, logic, and computer science
- Application of formal philosophy: against cognitivism, the study of the semiotic and then algebraic foundations of logic and computer science

### K.29.1.2 What is the purpose of this course?

In our "Information Age", characterized by an ever stronger interplay between conscious (humans) and unconscious (machines) communication agents, there is a growing necessity of a common background between philosophers and computer scientists. Think only, at the many ethical challenges of AI and of autonomous systems requiring a double and convergent competence between philosophers and computer scientists, before all for implementing the ability of performing autonomous ethical choices in machines, also in unpredictable situations. This is only one example of the many not only ethical, but also logical, epistemological and ontological issues related to the human-machine interface. Aim of the course is of presenting to the students some introductory notions of Formal Philosophy that, using Category Theory as the common algebraic metalanguage of the mathematical (extensional) and philosophical (intensional) logics ("inten$\underline{s}$ional" with "s" is the logic of the "inten$\underline{t}$ional" with "t" (conscious) states of mind), can suggest to young computer scientists a novel and intriguing perspective for approaching operationally old and new problems related to the artificial simulation of the intentional cognitive human behavior.

#### K.29.1.3 Course Objectives Based on Bloom's Taxonomy

**– What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define

- Philosophical or intensional logics as the logic of the intentional states of mind
- Intensional (ontic, epistemic, deontic) logics like as many semantics (models) of the same modal logical calculus
- Tarski's model theory in extensional logic *vs.* Kripke's model theory in modal and intensional logics
- The operator algebra approach in quantum physics and the the semantic notion of information in biological systems
- Category Theory as metalanguage of the operator algebra formalism in logic, mathematics, physics, and informatics

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- The difference between the "extensional" (mathematical) and the "intensional" (philosophical) logics
- Predication as "generalization" in extensional logic *vs.* predication as "specification" in intensional logic
- The extensional (algebraic, recursive, $\uparrow$) theory of predication as membership $\in$ in mathematical logic
- The intensional (co-algebraic, co-recursive, $\downarrow$) theory of predication as co-membership $\ni$ in philosophical logic
- The algebraic relational semantics of Tarski's Boolean algebras with operators in Category Theory logic
- The co-algebraic relational semantics of Kripke's modal Boolean algebra with operators in Category Theory logic
- The co-algebraic modeling of dissipative systems in quantum field theory, and its application in cognitive neuroscience and quantum computing

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply

- Different interpretations (ontic, epistemic, deontic) of the modal operators in different intensional contexts
- The distinction between extensional and intensional interpretations of predicates
- Some basic principles of Category Theory in the formalization of mathematical notions
- Some basic principles of Category Theory in the formalization of Tarski's Boolean logic with operators
- Some basic principles of Category Theory in the formalization of Kripke's modal Boolean logic with operators

#### K.29.1.4 Course evaluation

**Table K.69:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 0 |
| Interim performance assessment | 30 | 40 |
| Exams | 50 | 60 |

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

#### K.29.1.5 Grades range

**Table K.70:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table **??**, but it may change slightly (usually reduced) depending on how the semester progresses.

#### K.29.1.6 Resources and reference material


- Hansson, Sven Ove, Hendricks, Vincent F. (Eds.), *Introduction to formal philosophy*, Springer, Berlin-New York, 2018.
- Huges, G. E., Cresswell, M. J.; *A new introduction to modal logic*, Routledge, London, 1986
- Goranko, V., & Otto, M. "Model theory of modal logic". In P. Blackburn, F. J. van Benthem, & F. Wolter (Eds.), Handbook of Modal Logic. Amsterdam: Elsevier, 2007, pp. 225-331.
- Abramsky Samson and Tzevelekos Nikos, "Introduction to categories and categorical logic". In: *New structures for physics.* CLASS Notes in Physics, 813, B. Coecke, Ed., Berlin-New York, Springer, 2011, pp. 3-94
- Venema Y., "Algebras and Co-algebras". In: P. Blackburn, F. J. van Benthem, & F. Wolter (Eds.), Handbook of Modal Logic. Amsterdam: Elsevier, 2007, pp. 332-426
- Basti Gianfranco, "The quantum field theory (QFT) dual paradigm in fundamental physics and the semantic information content and measure in cognitive sciences", In: *Representation and Reality in Humans, Other Living Organisms and Intelligent Machines*, Dodig-Crnkovic, Gordana and Giovagnoli, Raffaela (Eds.), Springer International Publishing, Berlin-New York, 2017, pp. 177-210
- Freeman Walter and Vitiello Giuseppe, "Nonlinear brain dynamics as macroscopic manifestation of underlying many-body field dynamics", *Physics of Life Reviews,* vol. 3, no. 2, pp. 93-118, 2006
- Basti, Gianfranco, Capolupo, Antonio, and Vitiello, Giuseppe, "Quantum field theory and coalgebraic logic in theoretical computer science." *Progress in Biophysics and Molecular Biology* 130(2017), pp. 39-52

## K.29.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.71:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Formal philosophy in the history of philosophy | 29 |
| 2 | Formal philosophy of the natural and artificial computations | 29 |

### K.29.2.1 Section 1

**Section title:**

Formal philosophy in the history of philosophy

**Topics covered in this section:**

- The notion of formal philosophy as formalization of the philosophical doctrines using the axiomatic method
- Formal philosophy as a formal tool of the interdisciplinary dialogue between human and mathematical sciences
- Formal philosophy as an unavoidable first step toward the artificial simulation of human behaviors in AI
- Formal philosophy based on the distinction between mathematical (extensional) and philosophical (intensional) logics
- Mathematical logic based on the extensional interpretation of predication as membership ∈ (=generalization)ì
- Intensional logics based on the intensional interpretation of predication as co-membership ∋ (=context-dependent specification)
- The inten<u>s</u>ional logic (with "s") is the logic of the inten<u>t</u>ional (with "t") states of mind ("first person" statements: "*I/we* believe that...")
- In intensional logics, the "extensionality axiom" of the class logic and the "existential generalization axiom" of the (extensional) predicate logic do not hold.

- Intensional logics are "non truth-functional" logics, consisting in different interpretations (and then in different applied "truth criteria") of the two modal operators "necessity"/"possibility" $\Box/\Diamond$ of the modal calculus.
- The modal calculus adds to the propositional calculus some modal axioms, whose combination determines as many modal systems.
- Main intensional interpretations of modal axioms and operators:
- "Alethic", distinguishing logical and ontic (causal) necessity/possibility;
- "Deontic", distinguishing ethical and legal obligation/permission;
- "Epistemic", distinguishing between science/opinion, i.e., sound/unsound beliefs
- Related notions of "formal ontology" in formal philosophy and in computer science of semantic databases
- In computer sciences, formal ontology concerns the *different types of entities with their relations* for different groups, institutions, etc., and their *conceptual representations* in specific semantic databases (e.g., of the FDA, or of SEP, etc.), and then in semantic technologies in general.
  - Computational formal ontology is at the intersection among knowledge engineering, knowledge management, data mining, information retrieval, semantic web,...
- In formal philosophy, formal ontology is the theory of the pre-logical foundations of predication in the history of logic and of philosophy:
  1. "Nominalism": predicates are simply linguistic conventions (Sophist and Nihilist philosophers);
  2. "Conceptualism" (cognitivism): predicates are expressions of concepts in mind (Descartes, Leibniz, Kant);
  3. "Realism": foundation of predicates as independent of language and cognition in two senses:
     (a) "Logical": foundation of predicates in the realm of the pure logical forms (Plato, Frege, Gödel)
     (b) "Natural": foundation of predicates in the natural realm in two senses:
         i. "Atomism": only singular names denote individuals, so that predicates in logic denote their properties and relations in reality, with-

out any difference between them ("adjectival" predication: e.g., *"x* is white") and *natural kinds* ("nominal" predication: e.g., "$x$ is a horse") (Democritus, Wittengstein, Carnap);

ii. "Relational": supports the previous distinction. I.e., also common names denote, not individuals, but the *logical structures* of relations subject-predicates in logic are "mirroring" *causal structures* of natural kinds of things in nature (genera/species, e.g., "horses are mammalian" or individuals/species "Ribot is a horse") (Aristotle, Aquinas, Peirce, Kripke).

- In modern philosophy, the *relational natural realism* depends historically on the fundamental role of Peirce's semiotics, in overcoming modern conceptualism (cognitivism) in formal ontology, that persists in Frege's mathematical logic of classes.
- Peirce stated that logic it is not depending on human knowledge (=cognitivism), but it has an *algebraic foundation,* as far as we can endow algebraic relations and structures with an *intrinsic capacity of signifying.*
- Peirce vindicated – against Boole's and Schroeder's purely *dyadic* algebraic foundation of mathematical logic – the necessary *triadic* character of any *signifying* relation in the algebra of logic, i.e., the logic as *formal semiotics.*
- This led Peirce to the "invention" of the modern *algebra of relations,* as well as to the development of his famous *ante-predicative* triadic theory of the algebraic categories (*firstness, secondness, thirdness*) underlying any predicative theory of categories.
- Peirce's development of his pioneering algebraic theory of mathematical logic – contemporary to Frege's (predicative) mathematical logic of classes –, based on the usage of *commutative diagrams,* introduced for the first time in algebra by Peirce.
- This invention by Peirce of the algebra of relations remained not valorized till Tarski's *axiomatization* of the logic of relations into a *calculus of relations* in the second half of XX cent.
- This axiomatization includes also Peirce's *algebraic notion of category,* underlying the actual development of *Category Theory* as metalanguage of pure and

applied logic and mathematics – physics and computer science included –, which is wider than set-theory since, *sets and functions* category" **Set** is one of the possible algebraic categories.

- Each category indeed is intended as a *structure preserving* collection of 1) *morphisms* (arrows) – in the case of **Set**, functions –, 2) *objects* – in **Set** they are sets –, and 3) *compositions of morphisms.*

## What forms of evaluation were used to test students' performance in this section?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

### Typical questions for seminar classes (labs) within this section

1. The distinction between extensional (mathematical) and intensional (philosophical) logics in formal philosophy
2. Which are the two main axioms of the extensional (mathematical) logic that do not hold in intensional logic?
3. Which is the relationship between intensional logics and modal logic?
4. What means that the modal calculus is an extension of the ordinary propositional calculus?
5. Which are the main intensional interpretations (semantics) of the modal calculus?
6. Which is the distinction between formal ontology in computer science and in formal philosophy?
7. How can we use the formal ontology for synthesizing the main doctrines on the foundation of predication in the history of logic and philosophy?
8. Which is the relationship between the relational natural realism and Peirce's semiotics?
9. Which is the main contributions of Peirce to the history of algebra?

10. Which is the relationship between Peirce's theory of algebraic categories and the actual Category Theory?

#### K.29.2.2 Section 2

**Section title:**

Formal philosophy of the natural and artificial computations

**Topics covered in this section:**

- The beginning of the informational approach to biological sciences by
  1. N. Wiener's *Cybernetics* book
  2. Watson's and Crick's discovery of the *DNA double helix*
- The centrality of the notion of *signifying chemical signalling* among sub-systems of a living organism of whichever complexity as the "secret of life" in *biosemiotics*
- The relevance of an informational approach to biology emphasized by the discovery in *epigenetics* of the role of the inner/outer environment
- The relevance of the *environment*, the notion of living organism as a *dissipative system*, and the intrinsic *semantic* character of the *information processing* in biological systems
- The syntactic character of Shannon's measure of information and the issue of the *notion* and *measure* of semantic information
- The physical foundation of information in *quantum physics*
- Some basic notions of quantum mechanics (QM) mathematical formalism and its dependence on statistical mechanics
- The intrinsically *isolated* character of system representation in QM and the impossibility of representing in the QM formalism *phase transitions* of many-body physics (field theory)
- The related notion of Schrödinger wave function *decoherence* in QM if the system interacts
- The consequent purely *syntactic* (Shannon's) character of the "observer-related" notion and measure of information in QM
- The related notion of *Quantum Universal Turing Machine* in QM approach to quantum computing.

- The confusion between the statistical (Schrödinger) wave function and the physical wave function in quantum field theory (QFT), i.e., in quantum many-body-physics
- In the many-body representation of QFT (differently as to the standard (Dirac) representation of QFT like a "second quantization" as to QM) possibility of dealing with interacting and then *dissipative* quantum systems.
- Consistency of dissipative QFT with the *Third Principle of Thermodynamics* and its interpretation of the quantum vacuum (QV) as a thermodynamic *universal energy reservoir* with a temperature >0ºK
- The *Goldstone Theorem* as the core of QFT and the notion of the infinitely many "spontaneous symmetry breakings" (SSBs) of the QV, i.e., of the quantum fields at their ground state (>0ºK)
- Each SSB corresponds to the spontaneous establishment of *long-range correlations* or *phase coherences* of the quantum fields at their ground state.
- The notion of *quantum vacuum foliation,* i.e., the "piling-up" of quantum field phase coherences, each "dynamically labeled" by the fingerprint (= memory address) of a *unique condensate value of Nambu-Goldstone bosons,* as the quantum foundation of the notion of "complex systems" and of "memories" in biological systems
- Dissipative QFT as the fundamental physics of the *condensed matter physics,* the physics of biology included
- The *coalgebraic* $A \rightarrow A \times A$ mathematical formalism of the *doubling of the algebras,* and then of the *doubling of the Hilbert space* (system-thermal bath) for the "canonical (Hamiltonian) representation" of a dissipative quantum system in a far-from-equilibrium condition
- The consequent principle of the *doubling of the degrees of freedom* (DDF), using the "minimum of the free-energy" measure as a *dynamic* (observer-independent) criterion for defining the *statistical expectations* for the system (state) phase transitions.
- The related possibility of defining dynamically *a semantic measure* of information for quantum dissipative systems based on the minimum free-energy measure as *matching criterion system-environment* (= dynamic semantics)
- The related notion of the "doubled" and then *semantic qubit* in Quantum com-

puting based on dissipative QFT

- The "quantum brain" hypothesis based on the supposition that the quantum entanglement is the physical basis of the *biological consciousness*
- Penrose's and Hamroff's QM model of the quantum brain, based on the hypothesis that the "information gain" typical of the intelligent act is related with the Schrödinger wave function decoherence
- According to Hameroff's hypothesis this process is implemented the neuron microtubule quantum interactions where brain qubits could be processed
- The alternative hypothesis is by W. Freeman's and G. Vitiello's theory of the *dissipative brain*
- This hypothesis is based on the *personalistic* anthropology that the "self" subject of the intelligent and free acts is the *person*, neither her brain, nor her self-consciousness, using her brain as an organ
- Indeed a living brain is a *dissipative brain* exchanging with its environment energy and information
- Consistency of this hypothesis with the notion of *extended mind*, i.e., located not is some part of the brain but in the energy-information exchange brain-environment
- Logical-mathematical consistency of this hypothesis and of its *coalgebraic formalism* of the DDF with the *modal relational coalgebraic semantics* of Kripke models developed independently in computer science in the framework of Category Theory (algebraic) logic (Y. Venema, R. Otto, J. Rutten).
- Feeman's and Vitiello's hypothesis of using the QV-foliation as solution of the long-lasting problem of *long-term memory traces* in brain (natural "deep-learning").
- This comfirms Searle's hypothesis that the brain of an inten<u>t</u>ional subject is able to perform *inten<u>s</u>ional logic* calculations.
- Some historical notes on Category Theory (CT)
- Its dependence on Peirce's invention of the *algebra of relations* and *commutative diagrams*
- Peirce's work evolved into the axiomatization of the algebra of relations for defining a *calculus of relations* making of CT the proper metalanguage of the *operator algebra* formalism in any field of applied mathematics, quantum physics and theoretical computer science (TCS) before all.

- CT as *an arrow-theoretic way of thinking* since with respect to set-theory elements are not primitives in CT, they also are the domain-codomain of appropriate morphisms.
- Some basic notions of CT:
  - *Primitives*: morphisms, composition of morphisms, two maps assigning a domain and codomain to each arrow; *two axioms*: identity, associativity; *objects* as reflexive morphisms: $x = Id_x = \mathbf{1}_x$
  - *Category*: each structure-preserving collection of morphisms, compositions of morphisms, and objects are a category, e.g., **Set, Vect, Grp, Hilb, Pos, ...**; *functor*: a morphism mapping a category onto another one, or in a *covariant* (maintaining morphism directions and composition orders) or in a *contravariant* way (reversing directions and orders); *dual equivalence* between the categories of algebras $A \times A \to A$ **Alg** and coalgebras **Coalg** $A \to A \times A$ as far as linked by a contravariant functor $\Omega$ i.e., $\mathbf{Alg}(\Omega) \simeq \mathbf{Coalg}(\Omega^{\mathbf{op}})$.
- Some basic notions of CT logic relevant in TCS:
  - *Logical dual equivalence* between an assert $\alpha$ on a category $\mathscr{C}$ and its *dual* $\alpha^{op}$ on the category $\mathscr{C}^{op}$
  - *Stone representation theorem for Boolean algebras:* dual equivalence between the category of Boolean algebras and of the topological Stone spaces (sharing the same topology of the Hilbert space sub-algebras or $C^*$-algebras).
  - *Functorial dual equivalence* between the category of the Boolean Algebras with operators (BAO) (Tarski & Jonsson) and the category of the Coalgebras over Stone spaces: $\mathbf{BAO}(\Omega) \simeq \mathbf{SCoalg}(\Omega^{\mathbf{op}})$ in the framework of the operator algebra approach in TCS.
  - *Universal Coalgebra as General Theory of Systems* (J. Rutten). Possibility of defining the semantics of programs (Boolean algebras) on the physical states of the system as far as *coalgebraically modeled.*
  - Coalgebraic construction of the *Infinite State Black-Box Machine* $\mathbb{M}$ (J.Rutten, Y. Venema) as a particular implementation of an automaton as a *Labeled State Transition System*
  - On this basis it is possible to demonstrate that the QV-foliation (the long-

term memory of the dissipative brain) in QFT can be modeled as a quantum implementation of an *Infinite State Black-Box Machine* $\mathbb{M}$

- Indeed, the QFT coalgebras form one only category having as functor the *Bogoliubov Transform* for bosons (photons in condensed matter physics) as *state (phase) transition function* mapping a phase of condensate of photons onto the next one (Basti, Capolupo, & Vitiello)
- This construction is confirmed by the evidence that the Bogoliubov transform for bosons and then for photons is effectively a *hyperbolic tangent function,* i.e., the core of whichever *machine learning algorithm,* evidently not only in artificial neural networks and then in AI, but also in the *natural computation* of a dissipative brain, as Freeman and Vitiello suggested.
- In a word, formally, given that the topologies of the Stone spaces are the same of the algebras and coalgebras of quantum physics – the category of the $q$-deformed Hopf coalgebras of dissipative QFT included – the dual equivalence $\mathbf{BAO}(\Omega) \simeq \mathbf{SCoalg}(\Omega^{\mathbf{op}})$ can be naturally extended to the category of the $q$-deformed Hopf coalgebras for the contravariant application of the Bogoliubov Transform $T$ i.e., $\mathbf{BAO}(\mathbf{T}) \simeq \mathbf{SqCoalg}(\mathbf{T}^{\mathbf{op}})$

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for seminar classes (labs) within this section**

1. The origins of the information approach in biological sciences in Wiener's "Cybernetics"
2. Biosemiotics and chemical signaling in biological systems.
3. Epigenetics, living organisms as dissipative systems and the issue of semantic information in biology
4. Quantum mechanics statistical formalism and its difficulties in modeling dissipative systems in the foundations of condensed matter physics

5. Dissipative QFT, quantum vacuum and the III Principle of Thermodynamics
6. Dissipative QFT as the fundamental physics of biological systems
7. The coalgebraic modeling of dissipative QFT, and the doubling of algebras and of the Hilbert spaces
8. The semantic notion and measure of information in dissipative QFT
9. The "dissipative brain" and the "QV-foliation" as solution of the long-term memories problem in brain
10. The QV-foliation and its computatational modeling as "infinite state black-box machine"

## K.30 Introduction to Big Data

- **Course name:** Introduction to Big Data
- **Course number:** N/A

### K.30.1 Course Characteristics

#### K.30.1.1 Key concepts of the class

- Distributed data organization
- Distributed data processing

#### K.30.1.2 What is the purpose of this course?

Software systems are increasingly based on large amount of data that come from a wide range of sources (e.g., logs, sensors, user-generated content, etc.). However, data are useful only if it can be analyzed properly to extract meaningful information can be used (e.g., to take decisions, to make predictions, etc.). This course provides an overview of the state-of-the-art technologies, tools, architectures, and systems constituting the big data computing solutions landscape. Particular attention will be given to the Hadoop ecosystem that is widely adopted in the industry.

#### K.30.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

- The most common structures of distributed storage.
- Batch processing techniques
- Stream processing techniques
- Basic distributed data processing algorithms
- Basic tools to address specific processing needs

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to ...

- The basis of the CAP theorem

- The structure of the MapReduce
- How to process batch data
- How to process stream data
- The characteristics of a NoSQL database

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

- Use a NoSQL database
- Write a program for batch processing
- Write a program for stream processing

#### K.30.1.4 Course evaluation

**Table K.72:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 30 |
| Interim performance assessment | 30 | 30 |
| Exams | 50 | 40 |

If necessary, please indicate freely your course's features in terms of students' performance assessment.

#### K.30.1.5 Grades range

**Table K.73:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

If necessary, please indicate freely your course's grading features.

#### K.30.1.6 Resources and reference material

- Slides and material provided during the course.
- Vignesh Prajapati. Big Data Analytics with R and Hadoop. Packt Publishing, 2013
- Jules J. Berman. Principles of Big Data: Preparing, Sharing, and Analyzing Complex Information. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2013

### K.30.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.74:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction | 2 |
| 2 | Hadoop | 4 |
| 3 | HDFS | 4 |
| 4 | MapReduce | 4 |
| 5 | YARN | 4 |
| 6 | Optimizing Data Processing | 6 |
| 7 | Spark | 6 |

#### K.30.2.1 Section 1

**Section title:** Introduction

**Topics covered in this section:**

- What is Big Data
- Characteristics of Big Data
- Data Structures
- Types of Analytics

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Describe the 6 Vs
2. Describe the types of analytics

**Typical questions for seminar classes (labs) within this section**

1. Design the structure of a DB to address a specific analytics type
2. Give examples of the 6 Vs in real systems

**Test questions for final assessment in this section**

1. Design the structure of a DB to address a specific analytics type
2. Give examples of the 6 Vs in real systems

#### K.30.2.2 Section 2

**Section title:** Hadoop

**Topics covered in this section:**

- Data storage
- Clustering
- Design decisions
- Scaling
- Distributed systems
- The ecosystem

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Describe the Hadoop ecosystem
2. Structure of an Hadoop cluster
3. Describe the scaling techniques

**Typical questions for seminar classes (labs) within this section**

1. Configure a basic Hadoop node
2. Configure a basic Hadoop cluster

**Test questions for final assessment in this section**

1. Identify the Hadoop components useful to address a specific problem.
2. Configure an multi-node Hadoop system.

#### K.30.2.3 Section 3

**Section title:** HDFS

**Topics covered in this section:**

- Distributed storage
- Types of nodes
- Files and blocks
- Replication
- Memory usage

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Describe the characteristics of the different nodes
2. How files and blocks are managed
3. How memory is managed
4. How replication works

**Typical questions for seminar classes (labs) within this section**

1. Configure a HDFS cluster
2. Configure different replication approaches
3. Build a HDFS client
4. Use a HDFS command line

**Test questions for final assessment in this section**

1. Configure a HDFS cluster with some specific replication approaches
2. Build a HDFS client

#### K.30.2.4 Section 4

**Section title:** MapReduce

**Topics covered in this section:**

- Distributed processing
- MapReduce model
- Applications
- Tasks management
- Patterns

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Describe the MapReduce model
2. Describe tasks management
3. Describe patterns of usage

**Typical questions for seminar classes (labs) within this section**

1. Solve with MapReduce a specific problem
2. Implement a usage pattern

**Test questions for final assessment in this section**

1. Describe the advantages and disadvantages of the MapReduce model
2. Solve a task designing the solution using MapReduce
3. Solve a task designing the solution using a composition of usage patterns

### K.30.2.5 Section 5

**Section title:** YARN

**Topics covered in this section:**

- Resource manager
- Components
- Run an application
- Schedules

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Describe the resource manager
2. Describe the lifecycle of an application
3. Describe and compare the scheduling approaches

**Typical questions for seminar classes (labs) within this section**

1. Compare the performance of the different schedules in different load conditions
2. Configure YARN
3. Evaluate the overall performance of YARN

**Test questions for final assessment in this section**

1. Evaluate the performance of a specific configuration
2. Compare the different schedules

#### K.30.2.6 Section 6

**Section title:** Optimizing Data Processing

**Topics covered in this section:**

- CAP theorem
- Distributed storage and computation
- Batch Processing
- Stream Processing
- Usage patterns
- NoSQL databases

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Analyze the CAP theorem
2. Define the kinds of data storage available
3. Characteristics of batch processing
4. Characteristics of stream processing
5. Describe the usage patterns
6. Compare NoSQL databases

**Typical questions for seminar classes (labs) within this section**

1. Build a program to solve a problem with batch processing
2. Build a program to solve a problem with stream processing
3. Interact with a NoSQL database

**Test questions for final assessment in this section**

1. Identify problems and solutions related to the CAP theorem
2. Compare solutions with batch and stream processing approaches
3. Design a system using a NoSQL database

#### K.30.2.7 Section 7

**Section title:** Spark

**Topics covered in this section:**

- Architecture

- Use cases
- Job scheduling
- Data types
- SparkML
- GraphX

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Describe the architecture of Spark
2. Describe the types of schedulers
3. Different characteristics of the data types
4. Features of SparkML
5. Features of GraphX

**Typical questions for seminar classes (labs) within this section**

1. Analyze the performance of different schedulers
2. Write a program exploiting the features of each data type
3. Write a program using SparkML
4. Write a program using GraphX

**Test questions for final assessment in this section**

1. Compare the performance of different schedules with different loads
2. Extend the SparkML library with a custom algorithm
3. Extend the GraphX library with a custom algorithm

## K.31 Theoretical Mechanics

- **Course name:** Theoretical Mechanics
- **Course number:**

### K.31.1 Course Characteristics

### K.31.2 What subject area does your course (discipline) belong to?

Mechanics, mathematical modeling and calculating of mechanical systems.

#### K.31.2.1 Key concepts of the class

- Mechanics: Physical principles and methods for calculating kinematic, static and dynamic problems of mechanics

#### K.31.2.2 What is the purpose of this course?

Control of robots as mechanical systems requires knowledge of the physical principles and methods of mathematical description of the laws of motion. Robotics specialists should be able to describe the trajectories of objects in different spaces, calculate static, dynamic and shock loads on the system's bodies, investigate equilibrium conditions of systems, investigate vibrations, and be able to create mathematical models of mechanical systems.

The purpose of the course is to give basic and advanced knowledge on theoretical mechanics. The course covers kinematics of a particle and a rigid body, statics of rigid bodies, particle dynamics, dynamics of a system, analytical mechanics. The objective of the course is to give knowledge and skills which can be used further for calculating of kinematics, statics and dynamics of mechanical parts of robots and studying advanced courses on robotics.

#### K.31.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize

- Methods for describing the laws of motion of a particle and a solid,
- Methods for calculating the speeds and accelerations of points and bodies included in a mechanical system,
- Methods for studying the equilibrium of mechanical systems,
- Methods for creating differential equations of motion of a particle and a solid,
- Methods for creating differential equations of motion of a mechanical system based on the classical approach,
- Methods for creating differential equations of motion of a mechanical system based on methods of analytical mechanics.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- How to draw up and use calculation schemes,
- What calculation methods can be used to solve a specific problem,
- What calculation methods are appropriate to use when solving a specific problem,
- What limitations and errors are imposed by a specific method when solving a problem.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Analyze and explain mechanical phenomena based on the laws and theorems of theoretical mechanics,
- Apply the basic laws and methods of theoretical mechanics to solving technical problems,
- Create mathematical models, evaluate their value and the relativity of their limits of application.

#### K.31.2.4 Course evaluation

The course grades are given according to the following rules: Homework assignments (4) = 50 pts, Quizzes (5) = 10 pts, Midterm exam = 20 pts, Final exam 20 pts.

**Table K.75:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 10 |
| Interim performance assessment | 30 | 50 |
| Exams | 50 | 40 |

#### K.31.2.5 Grades range

**Table K.76:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

#### K.31.2.6 Resources and reference material

Main textbook:

- S. Targ Theoretical Mechanics. A short course, 1968
- D. Deleanu Theoretical mechanics. Theory and applications / Dumitru Deleanu – Constanta: Nautica, 2012
- Stephen T. Thornton and Jerry B. Marion Classical Dynamics of Particles and Systems. 5th edition, 2004

Other reference material:

- Meshchersky I.V. Collection of Problems in Theoretical Mechanics 2014
- Prof . Dr. Ing. Vasile Szolga Theoretical Mechanics, 2010
- S.M. Targ Kratki kurs teoreticheskoi mechaniki, 1986 - in Russian
- A.I. Lurie Analiticheskaya mechanika, 1961 - in Russian
- Sbornik kursovych rabot po teoreticheskoi mechanike. A.A.Yablonski, 2000 - in Russian
- Meshchersky I.V. Sbornik zadach po teoreticheskoi mechanike, 1986 - in Russian

### K.31.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.77:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Kinematics | 8 |
| 2 | Statics | 4 |
| 3 | Dynamics | 12 |
| 4 | Analytical mechanics | 4 |

#### K.31.3.1 Section 1

**Section title:** Kinematics

**Topics covered in this section:**

- Introduction to theoretical mechanics
- Kinematics of a particle
- Translatory and rotational motion of a rigid body
- Plane motion of a rigid body
- Spherical motion of a rigid body
- Motion of a free rigid body
- Resultant motion

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Calculate of the kinematic parameters of the particle according to the given laws of motion, it is required to determine:

- particle trajectory,
- particle velocity,
- particle acceleration and its normal and tangential components,
- radius of curvature of the trajectory.

2. Calculate of the kinematic parameters of the planar mechanism, determine:

    - velocity of specific points of the mechanism and angular velocity of the links of the mechanism using the method of instantaneous velocity centers,
    - velocity of specific points of the mechanism and angular velocity of the links of the mechanism using the analytical method,
    - acceleration of specific points of the mechanism and angular accelerations of the links of the mechanism.

3. Calculate of the kinematics of the complex motion of a point, determine:

    - transport, relative and absolute velocity of the point,
    - transport, relative, Coriolis and absolute acceleration of the point.

4. Calculate of the kinematics of gears, determine the gear ratio, angular velocities and angular accelerations of links, velocities and accelerations of specific points of links for:

    - gearbox with fixed axles,
    - planetary gearbox with parallel axes,
    - planetary gearbox with intersecting axes.

**Typical questions for seminar classes (labs) within this section**

1. Make a synthesis of the laws of motion of a point and a solid, taking into account given conditions and restrictions.
2. Do a kinematic analysis of complex planar mechanisms with a large number of links.
3. Do a kinematic analysis of complex planar mechanisms with several degrees of freedom.
4. Do a kinematic analysis of spatial mechanisms.
5. Do a kinematic analysis of the complex motion of a solid body.

**Test questions for final assessment in this section**

1. Describe the vector, coordinate, and natural methods of specifying particle motion. Show the transition from one method to another. Find the velocity and acceleration of the particle in various methods.
2. Define the angular velocity vector and the angular acceleration vector of the body. Prove the independence of these vectors from the choice of the pole. Use the Euler vector formula to find the velocities and accelerations of points of a rotating rigid body and a rigid body making plane motion.
3. Describe ways to set the orientation of a solid in space, including Euler angles, Tight-Brian angles, quaternions. Show the methodology for determining the angular velocity vector in these cases.
4. Show the methodology of kinematic analysis of planar mechanisms, including the method of composing the equations of motion for the points of the mechanism, the theorems on the velocities and accelerations of body points in plane motion, and the instantaneous center of velocity method.
5. Show the methodology of kinematic analysis of the complex motion of a particle, the theorems on the addition of velocities and accelerations for complex motion of a particle.

#### K.31.3.2 Section 2

**Section title:** Statics

**Topics covered in this section:**

- Basic concepts and principles of Statics
- Parallel forces and couples
- Equilibrium of a rigid body system in 2D
- Equilibrium of a rigid body system in 3D
- Friction
- Center of gravity

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Derive equilibrium equations for a system of concurrent forces.
2. Derive equilibrium equations for a solid in 2D.

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

3. Derive equilibrium equations for a system of two or more solids in 2D.
4. Derive equilibrium equations for a solid in 3D.

**Typical questions for seminar classes (labs) within this section**

1. Apply equilibrium equations to calculate the reactions of supports and forces in the rods of a truss in 2D.
2. Apply equilibrium equations to calculate the reactions of supports of a solid body in 2D.
3. Apply equilibrium equations for calculating the reactions of supports of a system of bodies in 2D.
4. Apply equilibrium equations to calculate the reactions of supports of a solid body in 3D.
5. Investigate the equilibrium of a system of bodies taking into account friction.

**Test questions for final assessment in this section**

1. Explain the basic axioms of statics.
2. Demonstrate the methods for determining the moment of force about a point and about an axis.
3. Demonstrate the methods for transformation a couple of forces.
4. Demonstrate the method for determining the principal vector and the principal moment of the force system.
5. Describe the method for transformation a force system to the simplest possible form.

### K.31.3.3 Section 3

**Section title:** Dynamics

**Topics covered in this section:**

- Particle dynamics
- Theorem of the motion of the center of mass of a system
- Theorem of the change in the linear momentum of a system
- Theorem of the change in the angular momentum of a system
- Some cases of rigid body motion.
- D'Alambert's principle
- Mechanical work and power
- Theorem of the change in the kinetic energy of a system
- The theory of impact
- Oscillations

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Derive and solve the differential equations of rectilinear and curvilinear motion of a particle.
2. Derive and solve differential equations based on the theorem on the motion of the center of mass of a system.
3. Derive and solve differential equations based on the theorem on the change in the angular momentum of a system.
4. Derive and solve the differential equations of rectilinear and curvilinear motion of bodies that form a system with one degree of freedom.
5. Derive and solve differential equations of motion based on the D'Alembert's principle.
6. Derive and solve the differential equation based on the theorem on the change in kinetic energy.

**Typical questions for seminar classes (labs) within this section**

1. Apply the differential equations of a particle to study the motion of a body in a field of gravity under the influence of air resistance.
2. Apply the differential equations of a particle to study oscillations.
3. Apply the theorem on the motion of the center of mass of a system to determine the dynamic reactions of the support of the mechanism.
4. Apply the theorem on the change in the angular momentum of a system to study the gyroscopic effect.
5. Apply the D'Alembert's principle to determine the dynamic reactions of the supports of a mechanical system
6. Apply the kinetic energy change theorem to determine the velosity of bodies of a mechanical system.

**Test questions for final assessment in this section**

1. Formulate the theorem of the motion of the center of mass of a system. Show for what problems this theorem is effective.
2. Formulate the D'Alambert's principle. Show for what problems the calculation method based on this principle is effective.
3. Describe the concept of force field. Show the method for determining the work of a force at a movement of a particle in a potential force field.
4. Describe the processes that occur upon impact and methods for calculating the law of motion of the body upon impact.

#### K.31.3.4 Section 4

**Section title:** Analytical mechanics

**Topics covered in this section:**

- Constraints and their classification
- Generalized coordinates
- Generalized forces
- The D'Alembert-Lagrange's principle
- The principle of virtual work

- The General Equation of dynamics
- Lagrange's equations
- The Hamilton's equations

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. What are generalized coordinates?
2. What are cyclic coordinates?
3. Derive the differential equations of a mechanical system based on the principle of virtual work
4. Derive the differential equations of a mechanical system based on the General Equation of dynamics
5. Derive the differential equations of a mechanical system based on Lagrange's equations

**Typical questions for seminar classes (labs) within this section**

1. Apply the principle of virtual work to study the laws of motion of a mechanical system with one degree of freedom.
2. Apply the General Equation of dynamics to study the laws of motion of a mechanical system with one degree of freedom.
3. Apply the Lagrange's equations to study the laws of motion of a mechanical system with several degrees of freedom.
4. Apply the Lagrange's equations to study the oscillations of a mechanical system with two degrees of freedom.

**Test questions for final assessment in this section**

1. Formulate the the principle of virtual work. Show for what problems the calculation method based on this principle is effective.
2. Formulate the the D'Alembert-Lagrange's principle. Show for problems tasks the calculation method based on this principle is effective.
3. Demonstrate methods for calculating generalized forces.
4. Show the different forms of writing the Lagrange equations and explain in which cases each of these forms will be more convenient.
5. Demonstrate the principles of choosing the most convenient method for solving a given specific problem of mechanics.

## K.32 Introduction to Robotics

- **Course name:** Introduction to Robotics
- **Course number:** R-01

### K.32.1 Course Characteristics

#### K.32.1.1 Key concepts of the class

Robotics; Robotic components; Robotic control.

#### K.32.1.2 What is the purpose of this course?

This course is an introduction to the field of robotics. It covers the fundamentals of kinematics, dynamics, and control of robot manipulators, robotic vision, and sensing. The course deals with forward and inverse kinematics of serial chain manipulators, the manipulator Jacobian, force relations, dynamics, and control. It presents elementary principles on proximity, tactile, and force sensing, vision sensors, camera calibration, stereo construction, and motion detection. The course concludes with current applications of robotics in active perception, medical robotics, autonomous vehicles, and other areas.

#### K.32.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to ...

- Model the kinematics of robotic systems.
- Compute end-effector position and orientation from joint angles of a robotic system.
- Compute the joint angles of a robotic system to reach the desired end-effector position and orientation.
- Compute the linear and angular velocities of the end-effector of a robotic system from the joint angle velocities.

- Convert a robot's workspace to its configuration space and represent obstacles in the configuration space.
- Compute valid path in a configuration space with motion planning algorithms.
- Apply the generated motion path to the robotic system to generate a proper motion trajectory.
- Apply the learned knowledge to several robotic systems: including robotic manipulators, humanoid robots.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to ...

- Name various applications of robots
- Describe the current and potential economic and societal impacts of robot technology
- Use the Jacobian to transform velocities and forces from joint space to operational space
- Determine the singularities of a robot manipulator
- Formulate the dynamic equations of a robot manipulator in joint space and in Cartesian space
- List the major design parameters for robot manipulators and mobile robots
- List the typical sensing and actuation methods used in robots
- Analyze the workspace of a robot manipulator
- List the special requirements of haptic devices and medical robots
- Effectively communicate research results

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

- Describe rigid body motions using positions, orientations, frames, and mappings
- Describe orientations using Euler angles, fixed angles, and quaternions
- Develop the forward kinematic equations for an articulated manipulator

- Describe the position and orientations of a robot in terms of joint space, Cartesian space, and operational space
- Develop the Jacobian for a specific manipulator
- Determine the singularities of a robot manipulator
- Write the dynamic equations of a robot manipulator using the Lagrangian Formulation
- Analyze the workspace of a robot manipulator

#### K.32.1.4 Course evaluation

**Table K.78:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Weekly quizzes | 20 | |
| Home assignments | 20 | |
| Project | 20 | |
| Midterm Exam | 20 | |
| Final Exam | 20 | |

#### K.32.1.5 Grades range

**Table K.79:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 92-100 | |
| B. Good | 80-91 | |
| C. Satisfactory | 65-79 | |
| D. Poor/Fail | 0-59 | |

#### K.32.1.6 Resources and reference material

- Siciliano, Sciavicco, Villani, and Oriolo, Robotics: Modeling, Planning and Control, Springe

## K.32.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.80:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction to robotics | 14 |
| 2 | Kinematics | 14 |
| 3 | Differential kinematics | 16 |
| 4 | Dynamics | 16 |

### K.32.2.1 Section 1

**Section title:** Introduction to robotics

**Topics covered in this section:**

- Introduction to Robotics, History of Robotics
- Introduction to Drones
- Introduction to Self driving cars
- Programming of Industrial Robot

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the difference between the manipulator arm and manipulator wrist
2. What is Node in ROS
3. What are the disadvantages of ROS
4. Write sensors which are used in self driving cars.
5. Describe the classical approach for deign self driving car

**Typical questions for seminar classes (labs) within this section**

1. Advantages and drawbacks of robotic manipulators
2. Programming industrial robots
3. Developing self driving car
4. Drones and controllers for them

**Test questions for final assessment in this section**

1. Typical commands for programming industrial manipulator motions
2. Types of robots and their application ares
3. Control of self driving car

#### K.32.2.2 Section 2

**Section title:** Kinematics

**Topics covered in this section:**

- Rigid body and Homogeneous transformation
- Direct Kinematics
- Inverse Kinematics

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Properties of Rotation Matrix
2. How to find Euler angles from rotation matrix
3. How to compute rotation matrix from knowing Euler angles
4. How to derive equations for direct kinematic problem
5. How to solve inverse kinematics problem

**Typical questions for seminar classes (labs) within this section**

1. Structure, properties, and advantages of Homogeneous transformation
2. Expression for rotation around an arbitrary axis
3. Euler angles
4. Difference between Joint and Operational spaces
5. Direct kinematics for serial kinematic chain
6. Piper approach for inverse kinematics

**Test questions for final assessment in this section**

1. Transformation between reference frames
2. Find Euler angles for given orientation matrix and transformation order
3. Transformation between Cartesian and operational spaces
4. Direct kinematic for SCARA robot
5. Inverse kinematic for SCARA robot

#### K.32.2.3 Section 3

**Section title:** Differential kinematics

**Topics covered in this section:**

- Differential kinematics
- Geometric calibration
- Trajectory Planning

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Write the matrix of differential transformation

2. What is Jacobian matrix
3. Difference between parametric and non-parametric robot calibration.
4. Why we need complete and irreducible model
5. How trajectory planning is realised
6. What is trajectory junction

**Typical questions for seminar classes (labs) within this section**

1. Jacobian matrix calculation
2. Jacobian matrices for typical serial manipulators
3. Robot calibration procedure
4. complete, irreducible geometric model
5. robot control strategies with offline errors compensation
6. Trajectory planning in joint and Cartesian spaces
7. Trajectory junction

**Test questions for final assessment in this section**

1. Write Jacobian for Polarrobot
2. Advantages and disadvantages parametric and non-parametric robot calibration.
3. complete, irreducible geometric model for spherical manipulator
4. Compute the joint trajectory q(t) from q(0) = 1 to q(2) = 4 with null initial and final velocities and accelerations. (polynomial)
5. Obtain manipulator trajectory for given manipulator kinematics, initial and final states and velocity and acceleration limits/

#### K.32.2.4 Section 4

**Section title:** Dynamics

**Topics covered in this section:**

- Dynamics of Rigid body
- Lagrange approach
- Newton-Euler approach

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Energy of rigid body
2. Dynamics of rigid body
3. What is Direct and Inverse Dynamics
4. Difference between Newton Euler and Lagrange Euler approaches

**Typical questions for seminar classes (labs) within this section**

1. Dynamics of rigid body
2. Direct and Inverse Dynamic
3. Newton-Euler Approach
4. Lagrange-Euler Approach

**Test questions for final assessment in this section**

1. Solve inverse dynamics problem for Cartesian robot
2. Solve direct dynamics problem for RRR spherical manipulator
3. Moving frame approach for dynamics modelling

# K.33 Fundamentals of Computer Security

- **Course name:** Fundamentals of Computer Security
- **Course number:**

## K.33.1 Course characteristics

### K.33.1.1 Key concepts of the class

- Basic security concepts
- Encryption techniques

### K.33.1.2 What is the purpose of this course?

This is an introductory course in computer security. During the course, students will learn the fundamental principles and techniques that constitute the basis for the modern computer security. The aim is to provide a practical knowledge of both the principles and practice of cryptography and network security. Practical applications that have been implemented and are in use to provide network security will be introduced in the final part of this course.

### K.33.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define

- Different security demands
- Basic Encryption techniques
- Symmetric encryption
- Asymmetrical encryption
- Message authentication algorithms
- Digital signature technique

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- Difference between security and cryptography
- Difference between stream cipher and block cipher
- Concept of DES and AES
- Symmetrical encryption
- Asymmetrical encryption
- Cryptographic hash function
- Difference between message authentication and Digital signature

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply

- How to encrypt and decrypt messages using different methods
- How to use Hash functions for cryptography
- Different brute force attacks

#### K.33.1.4 Course evaluation

**Table K.81:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 30 |
| Interim performance assessment | 30 | 10 |
| Exams | 50 | 60 |

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

#### K.33.1.5 Grades range

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table **??**, but it may change slightly (usually reduced) depending on how the semester progresses.

**Table K.82:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

#### K.33.1.6 Resources and reference material

- William Stallings, *Cryptography and Network Security: Principles and Practice, 6th Edition.*
- *Bruce Schneier, Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C (cloth).*

### K.33.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.83:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Classical Encryption Techniques | 8 |
| 2 | Symmetric encryption | 12 |
| 3 | Asymmetric encryption | 12 |
| 4 | Cryptographic hash functions | 8 |

#### K.33.2.1 Section 1

**Section title:**

Classical Encryption Techniques

**Topics covered in this section:**

- Present an overview of the main concepts of symmetric cryptography
- Explain the difference between cryptanalysis and brute-force attack
- The operation of a monoalphabetic substitution cipher
- The operation of a polyalphabetic cipher.

- The Hill cipher

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How many keys are required for two people to communicate via a cipher?
2. What is the difference between a block cipher and a stream cipher?
3. List and briefly define types of cryptanalytic attacks based on what is known to the attacker.
4. Briefly define the monoalphabetic cipher.
5. What is the difference between a monoalphabetic cipher and a polyalphabetic cipher?

**Typical questions for seminar classes (labs) within this section**

1. Using the two keys (memory words) cryptographic and network security, encrypt the following message.
2. Decrypt the ciphertext. Show your work.
3. Construct a Playfair matrix with the key largest.
4. Write a program that can encrypt and decrypt using the general Caesar cipher, also known as an additive cipher.
5. We have shown that the Hill cipher succumbs to a known plaintext attack if sufficient plaintext–ciphertext pairs are provided. It is even easier to solve the Hill cipher if a chosen plaintext attack can be mounted. Describe such an attack.

**Test questions for final assessment in this section**

1. What substitution system results when we use a 25 * 1 Playfair matrix?
2. Using the Vigenere cipher, encrypt the word "explanation" using the key leg.

3. How many one-to-one affine Caesar ciphers are there?
4. What is the difference between an unconditionally secure cipher and a computationally secure cipher?

#### K.33.2.2 Section 2

**Section title:**

Symmetric encryption

**Topics covered in this section:**

- Present an overview of the Feistel cipher and explain how decryption is the inverse of encryption
- Present an overview of Data Encryption Standard (DES)
- Discuss the cryptographic strength of DES
- Present an overview of the general structure of Advanced Encryption Standard (AES)
- Understand the four transformations used in AES

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the difference between the AES decryption algorithm and the equivalent inverse cipher?
2. How is the S-box constructed?
3. Briefly describe AddRoundKey.
4. What is the difference between ShiftRows and RotWord?

**Typical questions for seminar classes (labs) within this section**

1. Create software that can encrypt and decrypt using S-AES. Test data: A binary plaintext of 0110 1111 0110 1011 encrypted with a binary key of 1010 0111 0011 1011 should give a binary ciphertext of 0000 0111 0011 1000. Decryption should work correspondingly.
2. Implement a differential cryptanalysis attack on 1-round S-AES.
3. Given the plaintext 000102030405060708090A0B0C0D0E0F and the key 0101010101010101010101010101
4. Show the first eight words of the key expansion for a 128-bit key of all zeros.

**Test questions for final assessment in this section**

1. Compare AES to DES. For each of the following elements of DES, indicate the com- parable element in AES or explain why it is not needed in AES.
2. What was the final set of criteria used by NIST to evaluate candidate AES ciphers?
3. Verify the entry for 01 in the S-box.

### K.33.2.3 Section 3

**Section title:**

Asymmetric encryption

**Topics covered in this section:**

- An overview of the basic principles of public-key cryptosystems
- An overview of the RSA algorithm.
- Requirements for a public-key cryptosystem.
- The two distinct uses of public-key cryptosystems.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are the principal elements of a public-key cryptosystem?
2. What are the roles of the public and private key?
3. Whaat is single-linkage and what are its pros and cons?
4. In an RSA system, the public key of a given user is e = 31, n = 3599. What is the pri- vate key of this user? Hint: First use trial-and-error to determine p and q; then use the extended Euclidean algorithm to find the multiplicative inverse of 31 modulo f(n).

**Typical questions for seminar classes (labs) within this section**

1. In a public-key system using RSA, you intercept the ciphertext C = 10 sent to a user whose public key is e = 5, n = 35. What is the plaintext M?
2. Suppose we have a set of blocks encoded with the RSA algorithm and we don't have the private key. Assume n = pq, e is the public key. Suppose also someone tells us they know one of the plaintext blocks has a common factor with n. Does this help us in any way?

**Test questions for final assessment in this section**

1. In the RSA public-key encryption scheme, each user has a public key, e, and a private key, d. Suppose Bob leaks his private key. Rather than generating a new modulus, he decides to generate a new public and a new private key. Is this safe?
2. In using the RSA algorithm, if a small number of repeated encodings give back the plaintext, what is the likely cause?
3. Suppose we have a set of blocks encoded with the RSA algorithm and we don't have the private key. Assume n = pq, e is the public key. Suppose also someone tells us they know one of the plaintext blocks has a common factor with n. Does this help us in any way?

#### K.33.2.4 Section 4

**Section title:**

Cryptographic hash functions

**Topics covered in this section:**

- A hash function used for message authentication.
- The differences among preimage resistant, second preimage resistant, andcollision resistant properties.
- An overview of the basic structure of cryptographic hash functions.
- SHA-512.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What characteristics are needed in a secure hash function?
2. What is the difference between weak and strong collision resistance?
3. What is the role of a compression function in a hash function?
4. It is possible to use a hash function to construct a block cipher with a structure similar to DES. Because a hash function is one way and a block cipher must be reversible (to decrypt), how is it possible?

**Typical questions for seminar classes (labs) within this section**

1. Suppose H(m) is a collision-resistant hash function that maps a message of arbitrary bit length into an n-bit hash value. Is it true that, for all messages x, x' with $x \neq x'$, we have $H(x) \neq H(x')$ Explain your answer.
2. For SHA-512, show the equations for the values of W16, W17, W18, and W19.

**Test questions for final assessment in this section**

1. This problem introduces a hash function similar in spirit to SHA that operates on let- ters instead of binary data. It is called the toy tetragraph hash (tth).6 Given a message consisting of a sequence of letters, tth produces a hash value consisting

of four letters. First, tth divides the message into blocks of 16 letters, ignoring spaces, punctuation, and capitalization. If the message length is not divisible by 16, it is padded out with nulls. A four-number running total is maintained that starts out with the value (0, 0, 0, 0); this is input to the compression function for processing the first block. The com- pression function consists of two rounds.

2. Consider the SHA-3 option with a block size of 1024 bits and assume that each of the lanes in the first message block (P0) has at least one nonzero bit. To start, all of the lanes in the internal state matrix that correspond to the capacity portion of the initial state are all zeros. Show how long it will take before all of these lanes have at least one nonzero bit. Note: Ignore the permutation. That is, keep track of the original zero lanes even after they have changed position in the matrix.

## K.34 System and Network Administration

- Course name: System and Network Administration
- Course number: ?

### K.34.1 Course characteristics

#### K.34.1.1 Key concepts of the class

- Unix & system tuning
- Free and Open Source licensing
- Firmware & boot loaders
- Disks & partitioning
- Daemons' setup and operation
- Text processing tools & Regular Expressions
- Backup & Monitoring
- Network fundamentals
- Link aggregation
- DNS fundamentals
- Network discovery & sniffing
- Introduction to spoofing and man in the middle
- Cryptography from a practical perspective
- High Availability clusters
- Load-balancing clusters
- File-systems & shared-disk file-systems
- Storage clusters & distributed block devices
- Virtualization clusters
- Automated provisioning & configuration automation

#### K.34.1.2 What is the purpose of this course?

This course covers the system and network administration for GNU/Linux and BSD operating systems. The students with the minor of security and blockchain must be able to operate GNU/Linux as their main server-oriented system and perform different tasks such as setting up and operating various tools and daemons. This is a very essential course for the security expert which will give the students hands-on experience of system deployment, firmware, booting, partitions, volume management, system and network optimization. At the end of this course, the students will be equipped with the tools and skills that they can use in industrial production environments. This course is practice-oriented similarly to a certification training. After completion, the students could be entitled as (junior) system and network analysts.

#### K.34.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define

- Free and Open-Source Software paradigms
- Principles of the Unix culture
- Linux bonding modes and link aggregation
- CA vs intermediate vs leaf SSL certificates
- High Availability cluster architecture
- Load-balancing cluster architecture
- Virtualization architecture
- Distributed storage architecture
- Convergence and hyper-convergence

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- In-depth operational knowledge of GNU/Linux and system daemons
- How GIT differs from CVS
- How rather critical IT infrastructures are handled
- Difference between caching DNS forwarder and an authoritative DNS service
- Requirements and use-cases for High Availability
- Requirements and use-cases for load-balancing
- Requirements and use-cases for virtualization
- Requirements and use-cases for distributed storage

- Challenges & constraints with convergence and hyper-convergence
- Requirements to automate the deployments of several servers at once

**- What should a student be able to apply at the end of the course?**

- System & daemons troubleshooting - read the logs
- Install a GNU/Linux system and configure it for it to be used a server
- Fix the boot-loader or recover the root account
- Use Version Control Systems
- Build software daemons from source
- Build the Linux kernel from source
- Design the architecture of IT infrastructures
- Deploy and maintain IT infrastructures
- Network troubleshooting - check for open ports
- Setting up network link aggregations
- Securing SSL web browser sessions
- SSL cipher suites tuning
- Bootstrap guest machines from the host
- Deal with RAW and QCOW2 sparse files
- Evaluate the need for network disks (block devices)
- Evaluate the need for network file-systems versus shared-disk file-systems

#### K.34.1.4 Course evaluation

**Table K.84:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 70 |
| Interim performance assessment | 30 | 10 |
| Exams | 50 | 20 |

If necessary, please indicate freely your course's features in terms of students' performance assessment: The laboratory assessments are particularly taken care of, and the tasks do correspond with the teachings from the lectures. SNA laboratory assignments are plethoric hence there are multiple tasks to choose from every week, depending on students' skills and preferences.

#### K.34.1.5 Grades range

**Table K.85:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 90-100 |
| B. Good | 75-89 | 80-89 |
| C. Satisfactory | 60-74 | 70-79 |
| D. Poor | 0-59 | 0-69 |

If necessary, please indicate freely your course's grading features: The laboratory assignments are mandatory with a required minimum result of 6/10 - including retakes and late submissions - to complete the course. As a consequence, the grades are generally pretty high and therefore the grading ranges are scaled up.

#### K.34.1.6 Resources and reference material

- Joshua Davies, Implementing SSL / TLS Using Cryptography and PKI, Wiley Publishing, 2011
- Eric Raymond, The Cathedral & the Bazaar, O'Reilly Media, 2008
- Æleen Frisch, Essential System Administration, Third Edition, O'Reilly & Associates, 2002
- Evi Nemeth, Garth Snyder, Scott Seebass, and Trent R. Hein, UNIX System Administration Handbook, Third Edition, Prentice Hall, 2000
- Mark Sobell, A Practical Guide to the Unix System, Third Edition, Addison-Wesley, 1994
- GNU Manuals Online `https://www.gnu.org/manual/manual.html`
- The Linux Kernel documentation `https://www.kernel.org/doc/Documentation/`
- The Revised Slackware Book Project `https://www.slackbook.org/`

### K.34.2 Course Sections

#### K.34.2.1 Section 1

**Section title:**

Infrastructure fundamentals

**Topics covered in this section:**

- What is a server
- Unix basics
- Unix tips & tricks
- System preparation
- FOSS licensing paradigms
- Unix culture & traditions
- Daemons' setup and operation
- Software & kernel building
- Firwmare & boot loaders
- Disks & partitioning
- Text processing tools & Regular Expressions
- Shell scripting
- XML/CSS
- Backup & Monitoring
- Groupware, helpdesk & bug tracking

**What forms of evaluation were used to test students' performance in this section?**

Low-load weekly lab assignments (reports of two-three pages including command line outputs or screenshots)

**Typical questions for ongoing performance evaluation within this section**

- What is the difference between MIT, BSD and GPL licenses?
- How do those licensing models differ in terms of ethical and rhetorical goals?
- What characterizes the Unix system in terms of usability?

**Typical questions for seminar classes (labs) within this section**

- How to troubleshoot system and daemons?
- Install a GNU/Linux system and configure it for it to be used a server
- Fix the boot-loader or recover the root account
- Use Version Control Systems
- Build software daemons from source

- Build the Linux kernel from source
- Design the architecture of IT infrastructures
- Deploy and operate IT infrastructures e.g. a helpdesk and bug tracking engine

**Test questions for final assessment in this section**

- How to disable recent hardware mitigations in the Linux kernel?
- How to disable IPv6 at boot time?
- How GIT repositories does differ from CVS repositories in terms of architecture?
- What are the requirements to get a robust and spam-unfriendly Mail eXchange up and running?

#### K.34.2.2 Section 2

**Section title:**

Network & Security

**Topics covered in this section:**

- Network fundamentals & IPv6 intro
- Link aggregation
- NOC use-cases & DDoS mitigations
- DNS fundamentals
- SMTP fundamentals
- Network discovery & sniffing
- Spoofing intro
- Man in the middle intro
- Stripping SSL
- Stripping STARTTLS
- PKI intro & Let's encrypt
- SSL interception
- SSL pinning & certificate transparency
- Asymmetric ciphers from a practical perspective
- Key negotiation algorithms from a practical perspective
- Symmetric ciphers from a practical perspective
- Mode of operation & padding

**What forms of evaluation were used to test students' performance in this section?**

Low-load weekly lab assignments (reports of two-three pages including command line outputs or screenshots)

**Typical questions for ongoing performance evaluation within this section**

- Choose a Linux bonding mode and explain its pros/cons, then setup link aggregation with it
- Create a self-signed certificate, set it up against e.g. an HTTP service and use it from your browser
- Create a CA and sign a few certificates, see how your browser behaves compared to a self-signed certificate

**Typical questions for seminar classes (labs) within this section**

- How to troubleshoot network issues?
- How to setup network link aggregations?
- What are the requirements for an authoritative DNS service to be up and running?
- How to verify SSL certificates and sessions?
- How to tune SSL cipher suites for a service?

**Test questions for final assessment in this section**

- What is the difference between caching DNS forwarder and an authoritative DNS service?

#### K.34.2.3 Section 3

**Section title:**

Storage & clusters

**Topics covered in this section:**

- High Availability clusters
- Load-balancing clusters

- File-systems & shared-disk file-systems
- Storage clusters & distributed block devices
- Virtualization clusters
- Automated provisioning & configuration automation

**What forms of evaluation were used to test students' performance in this section?**

Low-load weekly lab assignments (reports of two-three pages including command line outputs or screenshots)

**Typical questions for ongoing performance evaluation within this section**

- How is a High Availability cluster architecture designed?
- How is a Load-balancing cluster architecture designed?
- How is a Virtualization architecture designed?
- How is a Distributed storage architecture designed?
- How are Convergence and hyper-convergence designed?

**Typical questions for seminar classes (labs) within this section**

- How to bootstrap guest machines from the host?
- How to deal with RAW and QCOW2 sparse files?
- How does network disks (block devices) compare to virtual disks as sparse files?
- How does network file-systems compare with shared-disk file-systems?

**Test questions for final assessment in this section**

- What are the architectural requirements and use-cases for High Availability?
- What are the architectural requirements and use-cases for load-balancing?
- What are the architectural requirements and use-cases for virtualization?
- What are the architectural requirements and use-cases for distributed storage?
- What are the challenges & constraints when attempting to define a convergent or even a hyper-convergent virtualization infrastructure setup?
- What are the requirements to automate the deployments of several servers at once?

## K.35 Information Retrieval

- **Course name:** Information Retrieval
- **Course number:** XYZ
- **Subject area:** Data Science

### K.35.1 Course characteristics

#### K.35.1.1 Key concepts of the class

- What is Information retrieval?
- How Information retrieval is being performed?

#### K.35.1.2 What is the purpose of this course?

The course is designed to prepare students to understand background theories of information retrieval systems and introduce different information retrieval systems. The course will focus on the evaluation and analysis of such systems as well as how they are implemented. Throughout the course, students will be involved in discussions, readings and assignments to experience real world systems. The technologies and algorithms covered in class includes machine learning, data mining, social network analysis, natural language processing and so on.

#### K.35.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

- What is information retrieval? What is its place in the modern digital world?
- What is the high-level architecture of information retrieval systems?
- How to evaluate search engines?

**- What should a student be able to understand at the end of the course?**

- Understand background theories behind information retrieval systems
- Become familiar with important techniques and best practices for analysis of information retrieval systems

- Understand the structure of the dictionary and postings
- Understand the key algorithms which are used to create and optimize dictionaries and postings

**- What should a student be able to apply at the end of the course?**

- Get hands-on experience of information retrieval systems development
- Be able to design and implement information retrieval system for a class of devices

#### K.35.1.4 Course evaluation

**Table K.86:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 50 | 50 |
| Interim performance assessment | 25 | 25 |
| Exams | 25 | 25 |

If necessary, please indicate freely your course's features in terms of students' performance assessment:

**Labs/seminar classes:**

- In-class participation 1 point for each individual contribution in a class but not more than 1 point a week (i.e. 14 points in total for 14 study weeks),
- overall course contribution (to accumulate extra-class activities valuable to the course progress, e.g. a short presentation, book review, very active in-class participation, etc.) up to 6 points.

**Interim performance assessment:**

- in-class tests up to 10 points for each test (i.e. up to 40 points in total for 2 theory and 2 practice tests),
- computational practicum assignment up to 10 points for each task (i.e. up to 30 points for 3 tasks).

**Exams:**

- mid-term exam up to 25 points,
- final examination up to 25 points.

**Overall score:** 100 points (100%).

#### K.35.1.5 Grades range

**Table K.87:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 85-100 | 85-100 |
| B. Good | 75-84 | 75-84 |
| C. Satisfactory | 60-75 | 60-75 |
| D. Poor | 0-59 | 0-59 |

If necessary, please indicate freely your course's grading features:

- A: more than 85 of the overall score;
- B: at least 85 of the overall score;
- C: at least 75 of the overall score;
- D: less than 60 of the overall score.

#### K.35.1.6 Resources and reference material

**Textbook:**

- Manning, Raghavan, Schütze, An Introduction to Information Retrieval, 2008, Cambridge University Press

**Reference material:**

- Baeza-Yates, Ribeiro-Neto, Modern Information Retrieval, 2011, Addison-Wesley
- Buttcher, Clarke, Cormack, Information Retrieval: Implementing and Evaluating Search Engines, 2010, MIT Press

## K.35.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.88:** Course Sections

| Section Number | Section Title | Lectures (hours) | Seminars (labs) | Self-study | Knowledge evaluation |
|---|---|---|---|---|---|
| 1 | Information retrieval basics | 12 | 6 | 12 | 2 |
| 2 | Scoring and evaluation | 8 | 4 | 8 | 1 |
| 3 | Relevance | 8 | 4 | 8 | 1 |
| 4 | Classification tasks | 12 | 6 | 12 | 2 |
| 5 | Clusterization | 8 | 4 | 8 | 1 |
| Final examination | | | | | 2 |

### K.35.2.1 Section 1

**Section title:** Information retrieval basics

**Topics covered in this section:**

- Introduction to IR.
- Boolean Model.
- The term vocabulary and postings lists.
- Dictionaries and tolerant retrieval.
- Index construction and compression

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How a text can be tokenized?
2. What are the most common techniques to process text before indexing?
3. What are the issues while defining a unit document?

**Typical questions for seminar classes (labs) within this section**

1. What is typical IR system architecture?
2. How term-document incidence matrix is organized?
3. Why do we call this data structure "index"? What is in common with the database index?

**Test questions for final assessment in this section**

1. What are the key operations of a Boolean query?
2. What is the purpose of positional indexes?
3. What are the important steps of the spellchecking algorithm?

#### K.35.2.2 Section 2

**Section title:** Scoring and evaluation

**Topics covered in this section:**

- Scoring, term weighting and the vector space model.
- Computing scores in a complete search system.
- Evaluation in information retrieval

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the definition precision and recall in IR?
2. Why do we need F1 metric, while there are recall and precision?
3. Provide an example, where we need relevance to be a real number but not a binary property.

**Typical questions for seminar classes (labs) within this section**

1. What is an nDCG formula?
2. How to estimate AP on the top 100 documents?
3. Why couldn't we calculate nDCG if we manually evaluated top 10 documents?

**Test questions for final assessment in this section**

1. When should we use AP and when mAP metric?
2. Explain each letter in nDCG.
3. What is a conceptual difference between nDCG and mAP? When one of them could be used and anther don't?

#### K.35.2.3 Section 3

**Section title:** Relevance

**Topics covered in this section:**

- Relevance feedback and query expansion.

- XML retrieval.
- Probabilistic information retrieval

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is an nDCG formula?
2. How to estimate AP on the top 100 documents?
3. Why couldn't we caclulate nDCG if we manually evaluated top 10 documents?

**Typical questions for seminar classes (labs) within this section**

1. Propose a strategy for A/B testing.
2. Propose recommender quality metric.
3. Implement DCG metric.
4. Discuss relevance metric.

**Test questions for final assessment in this section**

1. What is SBS (side-by-side) and how is it used in search engines?
2. Compare pFound with CTR and with DCG.
3. Explain how A/B testing works.

### K.35.2.4 Section 4

**Section title:** Classification tasks

**Topics covered in this section:**

- Language models for information Retrieval.
- Text classification and Naïve Bayes.
- Vector space classification.
- Support vector machines and machine learning on documents

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is a language model?
2. Why do we need to classify documents in vector space?
3. Provide an example, where additive smoothing is not adequate.

**Typical questions for seminar classes (labs) within this section**

1. White a formula for the naive
2. Why bigram models estimate joint probability better than the unigrams?
3. What is an additive smoothing?

**Test questions for final assessment in this section**

1. What is the difference between Naïve Bayes Classifier and Bayes Classifier for the bigram model?
2. Describe the Good-Turing smoothing method.
3. How to use SVM in documents classification?

#### K.35.2.5 Section 5

**Section title:** Clusterization

**Topics covered in this section:**

- Flat clustering.
- Hierarchical clustering.
- Matrix decompositions and latent semantic indexing.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. (i) Give an example of a set of points and three initial centroids (which need not be members of the set of points) for which 3-means converges to a clustering with an empty cluster. (ii) Can a clustering with an empty cluster be the global optimum with respect to RSS?
2. Make up a simple one-dimensional example (i.e. points on a line) with two clusters where the inexactness of cluster-based retrieval shows up. In your example, retrieving clusters close to the query should do worse than direct nearest neighbor search.
3. Suppose that $C$ is a binary term-document incidence matrix. What do the entries of $C^T C$ represent?

**Typical questions for seminar classes (labs) within this section**

1. Why are documents that do not use the same term for the concept car likely to end up in the same cluster in K-means clustering?

2. Show that K-means can be viewed as the limiting case of EM for Gaussian mixtures if variance is very small and all covariances are 0.
3. What is a difference between flat and hierarchical clustering?

**Test questions for final assessment in this section**

1. Define two documents as similar if they have at least two proper names like Clinton or Sarkozy in common. Give an example of an information need and two documents, for which the cluster hypothesis does not hold for this notion of similarity
2. For a fixed set of $N$ documents there are up to $N^2$ distinct similarities between clusters in single-link and complete-link clustering. How many distinct cluster similarities are there in GAAC and centroid clustering?
3. Suppose a run of HAC finds the clustering with K = 7 to have the highest value on some prechosen goodness measure of clustering. Have we found the highest-value clustering among all clusterings with K = 7?

# K.36 Digital Signal Processing

- **Course name:** Digital Signal Processing
- **Course number:** XYZ
- **Subject area:** Electric Engineering

## K.36.1 Course characteristics

### K.36.1.1 Key concepts of the class

- discrete(-time) signals, their impulse and frequency domains
- finite and infinite impulse response filters
- discrete(-time) Fourier transform and fast Fourier transform

### K.36.1.2 What is the purpose of this course?

The goal of the course is to present mathematical foundations of signal processing altogether with practical experience to design finite and infinite impulse response filters. The course is aimed to provide basic mathematical knowledge and practical skills needed for further studies of applied signal processing and digital signal processing from engineering as well as from mathematical perspective.

### K.36.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

- discrete time signals and systems, their classification,
- linear shift-invariant systems, filters and filtering,
- Discrete-time Fourier Transformation (DTFT),
- Discrete Fourier Transformation (DTFT),
- Fast Discrete Fourier Transformation (FDFT).

**- What should a student be able to understand at the end of the course?**

- relations between analog and digital signals,
- what are discrete signals, their convolution, auto-correlation and cross-correlation,

- role of impulse and frequency domains of discrete signals,
- differences between infinite and finite discrete signals,
- role of discrete time Fourier transform and its inverse,
- role of discrete Fourier transform (DFT) and fast DFT.

**- What should a student be able to apply at the end of the course?**

- basic numerical tools from mathematical package SciLab/Octave,
- classify discrete signals and systems,
- design and implement infinite and finite impulse response filters,
- implement and use discrete time Fourier transform,
- implement and use discrete Fourier transform and fast DFT.

#### K.36.1.4 Course evaluation

**Table K.89:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 20 |
| Interim performance assessment | 30 | 90 |
| Exams | 50 | 60 |

If necessary, please indicate freely your course's features in terms of students' performance assessment:

**Labs/seminar classes:**

- In-class participation 1 point for each individual contribution in a class but not more than 1 point a week (i.e. 14 points in total for 14 study weeks),
- overall course contribution (to accumulate extra-class activities valuable to the course progress, e.g. a short presentation, book review, very active in-class participation, etc.) up to 6 points.

**Interim performance assessment:**

- Each of 6 home-made individual lab (computational) assignments costs 15 points (i.e. 90 points for all 6 assignments).

**Exams:**

- in-class tests up to 10 points for each of 2 test (i.e. up to 20 points for both test),
- mid-term exam and final examination costs up to 20 points each (i.e. 40 points for both).

**Overall score:** 170 points (100%).

#### K.36.1.5 Grades range

**Table K.90:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 136-170 |
| B. Good | 75-89 | 119-135 |
| C. Satisfactory | 60-74 | 102-118 |
| D. Poor | 0-59 | 0-101 |

If necessary, please indicate freely your course's grading features:

- A: at least 80% of the overall score;
- B: at least 70% of the overall score;
- C: at least 60% of the overall score;
- D: less than 60% of the overall score.

#### K.36.1.6 Resources and reference material

**Textbook:**

- 

**Reference material:**

- 
- 

## K.36.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.91:** Course Sections

| Section Number | Section Title | Lectures (hours) | Seminars (labs) | Self-study | Knowledge evaluation |
|---|---|---|---|---|---|
| 1 | Complex numbers and functions, vector and Hilbert Spaces, computational aspects | 4 | 4 | 4 | 1 |
| 2 | Discrete-time signals and systems: properties and classification | 6 | 6 | 6 | 2 |
| 3 | Discrete-time Fourier Transformation | 6 | 6 | 6 | 0 |
| 4 | Discrete Fourier Transform (DFT) and Fast Discrete Fourier Transforms (FDFT's) | 6 | 6 | 6 | 1 |
| Final examination | | | | | 2 |

#### K.36.2.1 Section 1

**Section title:** Complex numbers and functions, vector and Hilbert Spaces, computational aspects

**Topics covered in this section:**

- Complex numbers and their matrix representation
- Vector spaces with dot-product
- Metrics and convergence, Hilbert spaces
- Algorithms and their computational (space and time) complexity

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Prove that each complex number has a square root.
2. Prove that the neutral element is unique in a vector space.
3. Prove that pixel (Manhattan) and Euclidean norms are equivalent in finite-dimensional real (complex) spaces.
4. Is the set of integers complete in the discrete metrics?
5. What is space and time complexity of dot product in a complex n-dimensional vector space?

**Typical questions for seminar classes (labs) within this section**

1. Prove that each complex number but zero has the inverse.
2. Prove that each vector of a vector space has unique opposite element.
3. Prove that pixel and the universal norms are equivalent in finite-dimensional real (complex) spaces.
4. Is the set of rational numbers complete in the discrete metrics?
5. What is space and time complexity of finite matrices multiplication (according to the definition)?

**Test questions for final assessment in this section**

1. Build if possible (or prove that it isn't) a real $2 \times 2$ matrix with given real eigen values $d > 0$ and $m > 0$.
2. Let $X$ be a set and function $d : X \to R$ be defined as $d(a, b) = \; if \; a = b \; then \, 0 \; else \, 10^6$. Prove that this function is a metrics.
3. What is time complexity to compute product of two real polynomials of order $n \geq 0$?.

#### K.36.2.2 Section 2

**Section title:** Discrete-time signals and systems: properties and classification

**Topics covered in this section:**

- Discrete signals as sequences, spaces $l^1$, $l^2$ and $l^\infty$

- Auto- and cross-correlation; memoryless, causal and shift-invariant systems
- Linear systems, their matrix representation and properties
- Convolution and its relations to linear shift-invariant systems

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 0 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Give examples of (infinite) signals in $l^1$, $l^2 \setminus l^1$, $l^\infty \setminus l^2$.
2. Is autocorrelation linear system? Is it shift-invariant?
3. Prove that a linear system is memoryless iff its matrix is diagonal.
4. Assuming a signal in $l^2$ and $\epsilon > 0$, evaluate number of component of the signal that is sufficient to compute autocorrelation with (component-wise) accuracy $\epsilon$.

**Typical questions for seminar classes (labs) within this section**

1. Prove that a linear system is causal iff its matrix is low-triangle.
2. A linear system is shift-invariant iff its matrix consists (exclusively) of diagonals of some constant (individual for each diagonal).
3. Assuming a signal $a$ in $l^2$, a signal $b$ in $l^2$ and $\epsilon > 0$, evaluate number of component of the signal $a$ that is sufficient to compute convolution $(a * b)$ with (component-wise) accuracy $\epsilon$.
4. Prove that product of finite power series is convolution of the finite signals consisting of the coefficients of these series.

**Test questions for final assessment in this section**

1. Compute cross-correlation of two box signals.
2. Study properties (linearity, causality, stability, etc.) of a weighted accumulator.
3. Prove associativity of convolution of finite signals (using finite series interpretation).

4. Let $u$ be Heaviside signal, $D$ — be signal delay operator, $\delta$ — be Kroneker signal, and $\mathbf{1}$ — be a constant signal where all components are 1; compare $(u * (\delta - D(\delta))) * \mathbf{1}$ and $u * ((\delta - D(\delta)) * \mathbf{1})$ and conclude about convolution associativity for the infinite signals.

#### K.36.2.3 Section 3

**Section title:** Discrete Fourier Transform (DFT) and Fast Discrete Fourier Transforms (FDFT's)

**Topics covered in this section:**

- Math preliminaries on complex exponent and Euler formulas.
- Introduction of the discrete-time Fourier transform via convolution eigen values and vectors.
- Discrete-time Fourier transform as the frequency response of a linear shift-invariant system.
- Inverse discrete-time Fourier transform.
- DTFT properties (including convolution theorem).
- Elements of ideal Filter Design.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Prove $e^a \times e^b = e^{a+b}$ for all complex numbers $a$ and $b$.
2. Do there exists a periodic function with periods 1 and $\sqrt{2}$ simultaneously?
3. Let $h$ be a signal. Express $H(e^{-j\omega})$ in terms of $H(e^{j\omega})$

4. Apply filter $h_n = \; if \; n = 0 \; then \; 0 \; else \; \frac{1}{n^2}$ to the signals with the angular frequencies $\omega \in \{0, \frac{\pi}{4}, \frac{\pi}{3}, \frac{2\pi}{3}, \frac{\pi}{2}, \frac{4\pi}{3}, \frac{5\pi}{4}, \pi\}$.
5. Prove conjugate property for DTFT: $x_n^* \overset{DTFT}{\longleftrightarrow} X^*(e^{j\omega})$.

**Typical questions for seminar classes (labs) within this section**

1. Assuming $h$ to be a real-valued signal; what are real-valued eigen-sequences of LSI-system $H$? (– I.e. what is $\omega$-frequency $\lambda_\omega$ to be real?).
2. Prove DTFT-correspondence for impulse shift: $x_{n-m} \overset{DTFT}{\longleftrightarrow} e^{-jm\omega} X(e^{j\omega})$.
3. Assume that $(c_n)$ is the cross-correlation of two signals $x$ and $y$; prove that $c_n \overset{DTFT}{\longleftrightarrow} X(e^{j\omega}) Y^*(e^{j\omega})$.
4. Prove that $\int_{-\infty}^{+\infty} f(t+t_0)\delta(t)dt = \int_{-\infty}^{+\infty} f(t)\delta(t-t_0)dt = f(t_0)$ (where $\delta$ is the Dirac Delta-function).
5. Design a low-band filter with a given spectrum consisting of a single box in the range $[-\pi, +\pi]$.

**Test questions for final assessment in this section**

1. Show that $e^{ix}$ is a periodic function, find the smallest period of $e^{ix}$.
2. Prove DTFT-correspondence for frequency shift: $e^{jn\phi} x_n \overset{DTFT}{\longleftrightarrow} X(e^{j(\omega-\phi)})$.
3. Prove sampling and scaling properties for the Dirac Delta function:
   - $g(t)\delta(t) = g(0)\delta(t)$;
   - $\delta\left(\frac{t}{a}\right) = |a|\delta(t)$ for every real $a \neq 0$.
4. Explain how to design a low-band filter with a given spectrum consisting of superposition of several boxes (each in the range $[-\pi, +\pi]$).

### K.36.2.4 Section 4

**Section title:** Discrete Fourier Transform (DFT) and Fast Discrete Fourier Transforms (FDFT's)

**Topics covered in this section:**

- Circular convolution and its relations to the linear convolution
- Eigen vectors and values of the circular convolution

- Discrete Fourier transform and its relations to DTFT
- Computational aspects of DFT and fast DFT
- Kotelnikov-Nyquist–Shannon theorem

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Then linear convolution of a circularly extended finite signal $x$ with period $m$ with any signal $h$ is also a signal with period $m$.
2. Let $u$ be the Heaviside filter and $h_n = \frac{u_n}{3^n}$. Compute and represent the periodized filters $h^1$, $h^2$, $h^3$, $h^4$ and $h^m$ (where $m \geq 5$).
3. Study commutativity, linearity and associativity of the circular convolution.
4. Compute and compare linear and circular convolution (with period $m \in [3..8]$) for signals $x = (\mathbf{4}, 5, 6, 2)$ and $y = (\mathbf{a}, 2, a)$.
5. Compute all circular exponent signals for $m = 4$.
6. Explain discrete Fourier transform as orthogonal vector decomposition.

**Typical questions for seminar classes (labs) within this section**

1. Prove that linear convolution of an absolutely summable filter with a circular signal equals to the circular extension of the periodized filter with the main period of the signal.
2. Prove circular impulse shift: $x_{(n-p) \mod m} \overset{DFT}{\longleftrightarrow} W_m^{kp} X_k$.
3. Give matrix representation for the circular convolution $(x^{(\ *\ )}\ y)$ where $x = (\mathbf{1}, 2, 3, 4, 5)$.
4. Recall 2-redex fast Fourier transform and draw matrices $A_m$ for several first natural values $m$.

**Test questions for final assessment in this section**

1. Assume that a finite signal $y$ has a finite support $[-m,+m]$ (for some $m \geq 0$); then $x_n *_n y_n = \sum_{k=-m}^{k=+m} x_k y_{n-k}$ every signal $x$ (not necessary finite).
2. Assume that signal $h \in l^1$; prove that periodized signal $(h^m)_n = \sum_{k\in\mathbf{Z}} h_{n+km}$ is correctly defined for all integer $m \geq 0$.
3. Prove circular frequency shift (also known as modulation): $W_m^{-kp} x_n \overset{DFT}{\longleftrightarrow} X_{(k-p) \mod m}$.
4. Give matrix representation for DFT and IDFT for some first natural $m$.
5. Explain the inverse of the discrete Fourier transform as reconstruction of a vector after its orthogonal decomposition.
6. Give example of applications of Kotelnikov-Nyquist–Shannon theorem.

## K.37 Data Mining

- **Course name:** Data Mining
- **Course number:** XYZ
- **Knowledge area:** Data Science

### K.37.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 3rd year of BS
- **Semester of instruction:** 2nd semester
- **No. of Credits:** 4 ECTS
- **Total workload on average:** 144 hours overall.
- **Class lecture hours:** 2 per week.
- **Class tutorial hours:** 0
- **Lab hours:** 2 per week.
- **Individual lab hours:** 2 per week
- **Frequency:** weekly throughout the semester.
- **Grading mode:** letters: A, B, C, D.

### K.37.2 Prerequisites

- Mathematical Analysis I
- Mathematical Analysis II
- Analytic Geometry and Linear Algebra I
- Analytic Geometry and Linear Algebra II
- Probability and Statistics
- Data Structures and Algorithms I
- Data Structures and Algorithms II
- Discrete Math and Logic

### K.37.3 Course outline

This course is designed for undergraduate students to provide core techniques of data processing and applications. Data Mining is an analytic process, which explores large data sets (also known as big data) to discover consistent patterns. This computational process involves a use of methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. This course will discuss advanced algorithms for classification, clustering, association analysis, and mining social network analysis. The subjects are treated both theoretically and practically through lab sessions.

### K.37.4 Expected learning outcomes

- Understand the entire chain of data processing
- Understand principle theories, models, tools and techniques
- Analyze and apply adequate models for new problems
- Understand new data mining tasks and provide solutions in different domains
- Design an appropriate model to cope with new requirements

### K.37.5 Expected acquired core competences

- Latest trends, algorithms, technologies in big data
- Ability to determine appropriate approaches towards new challenges
- Proficiency in data analysis and performance evaluations
- Application of models, combination of multiple approaches, adaptation to interdisciplinary fields

### K.37.6 Detailed topics covered in the course

- Foundations of interaction design
- Data Preprocessing
- Data Warehouse
- Association Rules
- Frequent Pattern mining

- Classification
- Clustering
- Recommendation Systems
- Mining graphs
- Mining data streams
- Neural Networks
- Outlier Detection
- Dimensionality Deduction

### K.37.7 Textbook

- Jiawei Han, Micheline Kamber and Jian Pei. *Data Mining: Concepts and Techniques (3nd Edition)*

### K.37.8 Reference material

- Jure Leskovec, Anand Rajaraman and Jeffrey D. Ullman. *Mining of Massive Datasets*

### K.37.9 Required computer resources

NA

### K.37.10 Evaluation

- Individual Assignments (30%)
- Course Project (20%)
- Mid-term Exam (20%)
- Final Exam (30%)

## K.38 Game Theory

- **Course name:** Game Theory
- **Course number:** XYZ
- **Knowledge area:** Intelligent Systems, Computational Sciences, Algorithms and Complexity

### K.38.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 3rd year of BS
- **Semester of instruction:** 2nd semester
- **No. of Credits:** 4 ECTS
- **Total workload on average:** 144 hours overall
- **Class lecture hours:** 2 per week.
- **Class tutorial hours:** 2 per week.
- **Lab hours:** 2 per week.
- **Individual lab hours:** 0.
- **Frequency:** weekly throughout the semester.
- **Grading mode:** letters: A, B, C, D.

### K.38.2 Prerequisites

- Analytic Geometry and Linear Algebra I
- Analytic Geometry and Linear Algebra II
- Probability and Statistics

### K.38.3 Course outline

This course is designed for undergraduate students to provide a comprehensive introduction to the main concepts and tools of game theory. Students will learn to represent an economic situation as a game and how to analyze it using different equilibrium concepts proposed in the literature, the prominent one being the Nash equilibrium.

Other topics include combinatorial games, zero-sum games, correlated equilibria, cooperative games, etc.

### K.38.4 Expected learning outcomes

This course will train you to apply machine learning methods for data science projects. After this course, you will be able to:

- Understand how to apply game theory to solve real-life problems
- Significant exposure to real-world applications of game theory
- To develop research interest in the theory and application of game theory

### K.38.5 Expected acquired core competences

- Core game theory algorithms
- Ability to model and solve real-world problems using game theory

### K.38.6 Detailed topics covered in the course

- Combinatorial games
- Zero-sum games
- General-sum games
- Nash equilibria
- Correlated equilibria
- Price of anarchy
- Voting
- Elicitation
- Scoring rules
- Adaptive decision making

### K.38.7 Textbook

- Fudenberg, Drew, and Jean Tirole. *Game Theory, MIT Press, 1991*

### K.38.8 Required computer resources

NA

### K.38.9 Evaluation

- Quizzes (20%)
- Labs (10%)
- Midterm Exam (25%)
- Final Exam (25%)
- Class and lab participation (10%)

## K.39 Project (SE, DS, AIR)

- **Course name:** Project (SE, DS, AIR)
- **Course number:** XYZ
- **Knowledge area:** Software Development Fundamentals and Social Issues and Professional Practice

### K.39.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 3rd year of BS
- **Semester of instruction:** 2nd semester
- **No. of Credits:** 4 ECTS
- **Total workload on average:** 144 hours overall
- **Class lecture hours:** 2 per week
- **Class tutorial hours:** 2 per week
- **Lab hours:** 2 per week
- **Individual lab hours:** 0
- **Frequency:** weekly throughout the semester
- **Grading mode:** letters: A, B, C, D

### K.39.2 Prerequisites

None

### K.39.3 Course outline

The primary goal of this course is for each student to pursue an independent project related to computer science. Each project will be pursued under the supervision of the supervisor of a student. A supervisor will be either a faculty member of Innopolis University or an industry partner. A project topic of a student will be decided through discussion with the student and his/her supervisor. Each student should carry out a project individually. However, it is allowed for two or three students work closely on similar topics, supervised by a common supervisor.

In this first part, students will be encouraged to concentrate on building software artifacts using knowledge they learned through their academic years.

### K.39.4 Expected learning outcomes

- Students will be able to develop medium to large sized projects on their own (under the supervision of supervisors).
- Students will be able to write reports about their projects effectively.
- Students will be able to present their projects effectively.

### K.39.5 Expected acquired core competences

- Ability to conduct a project independently
- Ability to design and build sizable software
- Ability to express project progress and results in a written and verbal form

### K.39.6 Textbook

No textbook is required.

### K.39.7 Reference material

NA

### K.39.8 Required computer resources

Students should have an access to computers.

### K.39.9 Evaluation

- Project Proposal (10%)
- Progress Report (10%)
- Final Report (30%)
- Final Project Artifacts (50%)

## K.40 Software Systems Design

- **Course name:** Software Systems Design
- **Course number:** XYZ
- **Knowledge area:** Software Development Fundamentals, Software Engineering, Systems Fundamentals

### K.40.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 3rd year of BS
- **Semester of instruction:** 1st semester
- **No. of Credits:** 4 ECTS
- **Total workload on average:** 144 hours overall
- **Class lecture hours:** 2 per week
- **Class tutorial hours:** 2 per week
- **Lab hours:** 2 per week
- **Individual lab hours:** 0
- **Frequency:** weekly throughout the semester
- **Grading mode:** letters: A, B, C, D

### K.40.2 Prerequisites

- Data Structures and Algorithms I
- Data Structures and Algorithms II
- Discrete Math/Logic
- Introduction to Programming I
- Introduction to Programming II

### K.40.3 Course outline

The need for building large scale Software has been increased in the last two decades. Nowadays, Software engineers are less likely to design data structures and algorithms from scratch. They are required to build systems from library and framework components. The Software System Design course aims to teach students the m ain concepts related to the construction of software systems. The course covers technical topics such as concepts of design for complex systems, object oriented programming, UML notation, among others.

### K.40.4 Expected learning outcomes

After taking the course, students will

- understand and apply object-oriented design techniques;
- develop and evaluate software systems;
- express the specifications and design of an application using UML;
- specify parts of the design using a formal design language (OCL);
- have experience designing medium-scale systems with patterns;
- have experience testing and analysing software.

### K.40.5 Expected acquired core competences

- Formal Models and Semantics
- Requirements Engineering
- Software Design
- Software Engineering
- Software Evolution

### K.40.6 Textbook

- Grady Booch, Robert Maksimchuk, Michael Engle, Bobbi Young, Jim Conallen, and Kelli Houston. 2007. Object-Oriented Analysis and Design with Applications, Third Edition (Third ed.). Addison-Wesley Professional.
- Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado, UML 2 Toolkit, OMG Press, 2004

### K.40.7 Reference material

- Lecturing and lab slides and material will be provided

### K.40.8 Required computer resources

Students should have laptops with basic software for reading and editing document.

### K.40.9 Evaluation

- Course Project (30%)
- Mid-term Exam (30%)
- Final Exam (30%)
- Class and lab participation (10%)

# K.41 Lean Software Development

- **Course name:** Lean Software Development
- **Course number:** XYZ

## K.41.1 Course characteristics

### K.41.1.1 Key concepts of the class

- Fundamental principles of producing software as a creative act of the human mind
- Techniques to optimize such production, with specific focus on agile methods

### K.41.1.2 What is the purpose of this course?

This course exposes the student to the core concepts behind Lean Development in Software Engineering, beyond myths and legends, emphasizing how it relates to the general principles of Lean Development. It discusses the different possible software processes, how they can be tailored, enacted, and measured. In addition, a significant part of the course is centered around the application of lean to software development to knowledge intensive areas not necessarily connected to software.

### K.41.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define:

- creative nature of software production as an act of creativity of the human mind,
- the substantial differences between tame and wicked problems,
- the core concepts of measurement in software engineering,
- the fundamentals of Taylorist/Fordist approaches to (software) production,
- the basis of lean and agile software development,
- the "dark" side of agility,
- the importance of knowledge and knowledge sharing in producing software,
- how to create an ad-hoc process for a development organization.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples):

- when a problem is “easy to solve” provided enough effort is put to such solution and when it is not,
- what is a measure in general,
- why it is important and how we can define and perform measurements in software engineering, especially in lean and agile development environments,
- how to organize the development process to collect metrics non invasively,
- the difference between pulling and pushing in (software) development,
- the fundamental principle of agility,
- the risk intrinsic in the dark side of agile,
- how to organize an agile development process based on the definition of overall Goals, associated Question, and milestones based on Metrics,
- an environment based on experience, like the Experience Factory.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to:

- compute the fundamental software metrics to track the evolution of a project,
- organize the aims of a (software) development organization in terms of Goals, Questions, and Metrics,
- create a tailored (lean and agile) development process for an organization producing software,
- define a path to insert and manage such (lean and agile) development process into an organization producing software,
- structure the experience gathering during inside an organization to based on it the future strategic decision of such organization,
- relate the various proposals for Agile Methods to the overall principles of Lean Management,
- define a suitable (lean) process for a new organization, a process to introduce

and institutionalize it, and an approach to measure the outcome of such introduction and institutionalization.

#### K.41.1.4 Course evaluation

**Table K.92:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes (weekly evaluations) | 20 | 35 [a] |
| Interim performance assessment | 30 | (merged with the above) |
| Exams | 50 | 65 [b] |

[a]Please note that Labs/seminar classes (weekly evaluations) is divided in 10% for class participation and weekly tests, 10% for the Weekly GQM and 15% for invited speaker presentations, questions, and answers.

[b]Of which 25% for the report on the application of Lean Principles to artistic areas, 10% for the overall GQM, and 30% for the oral.

Each component apart from the weekly tests will be assessed on a scale 0-10, where 6 is the minimum passing grade. In case of exceptional work a 10 cum laude will be assigned, with a numeric value from 11 to 13 at discretion of the instructor. The homeworks will be initially graded on a scale 0-2 weekly and then the overall grade will be assembled on a scale 0-10.

The grading, though, is not a simple linear combination of the components above. In particular:

- failing any part of the evaluation will trigger a failure in the entire course,
- the weekly quizzes will be averaged on the (n-4) best performances, where the 4 will be used to consider any kind of absence of the student, without requiring any additional documentation; if the student has legitimate reasons to be absent more than 4 times, then supporting document should be provided for all her/his absences,
- if there are not failing components, the final grade will be computed as a weighted average of the components above approximated at the highest second digit and then rounded to the closest integer.

Note that the questions for the exam are taken from the textbook.

#### K.41.1.5 Grades range

**Table K.93:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 96-100 |
| B. Good | 75-89 | 66-95 |
| C. Satisfactory | 60-74 | 56-65 |
| D. Poor | 0-59 | 0-55 |

The semester starts with the default range as proposed in the Table K.93, but it may change slightly (usually reduced) depending on how the semester progresses. See above.

#### K.41.1.6 Resources and reference material

- **Textbook:**
- **Reference:**
- **Reference:**
- **Reference:**
- **Reference:**
- **Reference:**
- **Reference:**

### K.41.2 Course Sections

The course is organized in 15 weeks with every weeks 2 academics hours of lectures and 4 academic hours of labs. The main sections of the course and approximate hour distribution between them is as follows:

#### K.41.2.1 Section 1

**Section title:**

Software as a creative activity

**Table K.94:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Software as a creative activity | 12 |
| 2 | Measurement in software | 24 |
| 3 | Taylorism and Fordism | 6 |
| 4 | Lean and Agile | 24 |
| 5 | Issues in Lean and Agile | 6 |
| 6 | Structuring a Lean Approach to software development | 6 |
| 7 | Optimizing the development process | 12 |

### Topics covered in this section:

- Nature of software
- Software and art
- Core resources for the production of software
- Tame and wicked projects
- Organizing the activities based on the GQM approach

### What forms of evaluation were used to test students' performance in this section?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

### Typical questions for ongoing performance evaluation within this section

1. Provide examples of creativity in the production of software.
2. Describe the differences between tame and wicked projects.
3. Discuss the key resources needed for the production of software.
4. What are key issues in creative production of software for distributed teams?

### Typical questions for seminar classes (labs) within this section

1. Provide examples of wicked problems from your everyday life
2. Evidence wickedness in different aspects of software production
3. Create a GQM for your aims of the semester
4. Discuss the role of GQM in tame and wicked projects

**Test questions for final assessment in this section**

1. Present the salient aspects of wicked problems
2. List the key aspects of software that make it a wicked problem
3. Provide a link between wickedness and creativity in software production
4. Discuss what promotes and what inhibit creativity in general and in software production
5. Outline meaning and limitation of the concept of "engineering" the production of software

#### K.41.2.2 Section 2

**Section title:**

Measurement in software

**Topics covered in this section:**

- Meaning of measures
- The representational theory of measurement
- Measurement scales
- Fundamental measures for the production of software
- Procedural measures
- Object Oriented measures

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Provide examples of the representational theory of measurement
2. List the measurement scales
3. Present for each measurement scale the operations that can be performed on it

4. Discuss the representational condition
5. What are key size metrics?
6. What are key complexity metrics?

**Typical questions for seminar classes (labs) within this section**

1. Provide examples of subjective and objective metrics
2. List 3 direct and 3 indirect measures, evidencing also the problems connected to the construction of indirect measures
3. Compute LOC and other size metrics for code snippets
4. Compute MCC and other complexity metrics for code snippets
5. Compute the metrics of the CK suite for portions of Object Oriented systems

**Test questions for final assessment in this section**

1. Structure the aims of a company using the GQM and detailing the metrics to compute, also explaining the deductions and the predictions that can be made with such metrics.
2. Given an index computed as a combination of metrics, determine if it is a metrics according to the representational theory of measurement.
3. Analyse a portion of a system and determine suitable metrics to extract and the information that would be provided by such metrics.
4. Given a website performing a service (like flight reservations), compute the Function Points for such website.
5. Discuss how to structure a taxonomy of quality for a specific company, explaining the role of reliability in it, and detailing how to compute the reliability again in the context of such company; please make the assumptions that you need to perform such computation.

#### K.41.2.3 Section 3

**Section title:**

Taylorism and Fordism

**Topics covered in this section:**

- The increase of productivity in the idea of Taylor
- The role of division of work
- Planning and formalization of tasks
- Economies of scale
- Problems in understanding tasks
- Taylorism/Fordism and software development

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Detail the fundamental assumptions by Taylor
2. What are the fundamental activities of managers according to Taylor?
3. In which sense Taylor has influenced what we now consider "good management practices?"
4. How does creativity relates to the "good management practices" of Taylor?
5. What are the problems in applying Fordism/Taylorism to software development?

**Typical questions for seminar classes (labs) within this section**

1. Provide examples of companies where the approach by Taylor has been successful
2. Discuss how the approach of Taylor can be useful in attracting and/or retaining employees
3. Analyse the fundamental activities of managers according to Taylor and determine their limits
4. Determine the fundamental activities of managers according to Taylor can have an implication in the software crisis
5. Outline how flexibility and variable requirements can be handled in the context of Taylorism and Fordism

**Test questions for final assessment in this section**

1. What aspects of making software feel like an art, which like a craft to you and how could you bend Taylorism and Fordism to handle it?
2. How does Fordism and Taylorism can explain that companies producing software containing bugs can still stay in the market, and, in certain case, also be successful?

3. Based on your experience and previous courses, which development models can refer to Fordism and Taylorism and which cannot be reduced to it?
4. How does the specialization of work in software development can be linked to Fordism and Taylorism?
5. Provide an example for each of the fundamental activities of managers according to Taylor that shows how such activity is very useful in software development and one that shows that is is inadequate.

#### K.41.2.4 Section 4

**Section title:**

Lean and Agile

**Topics covered in this section:**

- Taiichi Ono and the Toyota Production System
- Creating a "Radiography" of the Production Process
- Workers involvement
- "Pull" and Not "Push"
- Kanban
- Quality management
- Process control
- Job enrichment
- Control and coordination mechanisms
- Case study: Extreme Programming

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Present the key problems in batch production

2. Outline the key principles of the approach of Ono at Toyota
3. What are the fundamental steps in eliminating waste according to Ono?
4. Details the role of the customers and of the workers in the approach of Ono
5. Explain the difference between “Pulling” and “Pushing”
6. What are key steps in improving quality according to Ono?
7. What are the key control and coordination mechanisms available?

**Typical questions for seminar classes (labs) within this section**

1. What are the fundamental two actions needed to perform a Lean transformation according to Ono?
2. What are the associated three major needs?
3. What are the 5 steps to enact Lean Thinking according to Womak and Jones?
4. Discuss the 8 constantly ongoing activities in a lean company like Toyota
5. How can activities been classified in a decision matrix in an environment like Toyota?
6. Provide concrete examples of “Push” and of “Pull” in software production.
7. Details the control and coordination mechanisms present in agile and in traditional development environments.

**Test questions for final assessment in this section**

1. Structure a model like PDSA for a company producing websites for online marketing.
2. How can the right part and the right information be always available without waste according to the Toyota approach?
3. Compare a street crossing based on traffic lights with a roundabout and determine the approach that is safest and the one with the a highest throughput according to Waterfall and to Lean.
4. Discuss the involvements of workers in tayloristic/fordistic and in Lean development processes and their implications for the retention and the improvement of the quality of the workforce.
5. Outline the extent of which economies of scale exist in Lean development processes.

6. Imagine you had to introduce Extreme Programming in a software development team that follows a waterfall process. Which problems do you foresee? How will the clients react (that until now are used to work with a team that used the waterfall process)? How would you address them?

#### K.41.2.5 Section 5

**Section title:**

Issues in Lean and Agile

**Topics covered in this section:**

- The "Hype of Agile"
- The dark side of agile
- Skepticism about agile methods
- Knowledge and software engineering
- Using burn-down charts
- The Zen of agile

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the Gartner's Innovation Hype Cycle? Could you provide examples of application of it to a different field of knowledge?
2. Describe the so-called "Dark Agile Manifesto"
3. What are the sources of the skepticism present with respect to Agile?
4. What makes agile awkward in the eyes of "traditional" managers?
5. Given a production system, how do you determine what is the value of every step, how much improvement can be considered enough, and when is the point reached where value is not increased anymore but destroyed?

6. How is it possible to obtain knowledge about the production process? How can I create visibility of the ongoing activities or problems?

**Typical questions for seminar classes (labs) within this section**

1. Identify in other areas of software engineering phenomenon similar to the "Dark Side of Agile."
2. Identify in other knowledge-intensive fields phenomena similar to the "Dark Side of Agile" and discuss how they can be tackled.
3. Given a production process, determine strategies to store knowledge to create experience?
4. How can you design the production process so that the team uses the gained experience?
5. How can you systematically improve your process, also building on the experience to anticipate problems (create wisdom)?

**Test questions for final assessment in this section**

1. Imagine you are the boss of a small software development company. Which actions would you do or which practices would you introduce to prevent that your programmers fall into the trap of following a software guru?
2. Elaborate possible extreme (and damaging) positions that can be taken by gurus of agile.
3. Explain the differences in introducing a methodology by a guru and by a smart and effective coach.
4. Detail why Extreme Programming produces an informative workspace.
5. Discuss effective ways of packaging and "distributing" knowledge in software teams, starting with the guru approach of organizing knowledge into simple, clear practices which are easy to explain and to follow.

### K.41.2.6 Section 6

**Section title:**

Structuring a Lean Approach to software development

**Topics covered in this section:**

- Existing proposals to create a "Lean Software Development"
- Sharing a common vision
- Depriving gurus of their power
- GQM+
- Applying the GQM+ step-by-step
- Business alignment
- GQM+ for business alignment

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Reflect on the seven principles for characterizing Lean Software Development by Mary and Tom Poppendieck.
2. What is the semantic gap and which threats it poses to effective software development?
3. Map the structure of Extreme Programming to the layered structure of Shalloway et al.
4. How does the scientific method deprives gurus from their power?
5. What are the steps to implement the GQM+?
6. For what reason measurement goals and business goals should be interconnected?

**Typical questions for seminar classes (labs) within this section**

1. How do the practices of Lean Management defined by Hibbs and colleagues relate to the seven principles by Mary and Tom Poppendieck
2. Propose how you could develop software and hardware tools to promote common visions in companies.

3. Elaborate a proposal to create some kind of Balanced Scorecards to evaluate your current study.
4. Discuss the opinion of Ono about following plans and the extent to which such opinion contradicts (a) the practices, and (b) the principles of the tayloristic/fordistic approach
5. Propose a SWAT analysis to introduce a Lean approach to your most recent software development endavour

**Test questions for final assessment in this section**

1. Discuss the role of customer on-site under the perspective of lean and outline it relevance in the earlier and then in the later proposals of agile software development.
2. Suppose that you have to develop Balanced Scorecards for a software development team. Which perspectives would you use? Which goals would you use for each perspective?
3. What is a socio-technical system and how can it be used to describe an (agile) software production environement?
4. Imagine you want to evaluate how readable the source code of some program is. Define a GQM+ model to describe what and why you would measure.
5. The development in company M occurs according to the following schema: when a new project is started, a developer takes an old project that is the most similar to the new requirements and makes a copy and starts implementing the required modifications. To improve this process and to help the company to adopt a component-based approach, we want to understand which pieces of code are the best candidates for future components and which variability points they have. Define a suitable GQM+ for such purpose.

#### K.41.2.7 Section 7

**Section title:**

Optimizing the development process

**Topics covered in this section:**

- Why the PDSA does not work in software
- The experience factory
- The QIP cycle
- Non invasive measurement
- The big-brother effect
- The role of autonomation
- Employing Andon boards

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Details the major components of an experience factory.
2. What are Reflection, Retrospective, and Post-Mortem Analysis? Why are they useful in Lean Software Development?
3. What are the key components of a non invasive software measurement systems?
4. What is the big brother effect and how it is possible to alleviate it.
5. Discuss how autonomation is present in Extreme Programming.
6. Where is the term "Dashboard" coming from and what is its use in Lean Software Development?

**Typical questions for seminar classes (labs) within this section**

1. List the steps of a QIP.
2. Discuss the risks of a measurement program and how non invasive software measurement can help alleviating them.
3. Are there cases in which Theory X of management could be more effective than Theory Y? Discuss your findings.
4. For which aspects of software production autonomation could be useful?

5. Which tools could be used to promote autonomation?
6. Prototype by paper and pencil an Andon board that you would consider useful in a software production environment.

**Test questions for final assessment in this section**

1. What type of wisdom (in the sense of “know-why”) would you manage in an Experience Factory to support Lean Thinking? Distinguish between organizational learning and project learning.
2. Assume you are a manager convinced that Theory X is true. Which non-invasive measurement probes would you want to develop to maximize productivity? Now assume you are convinced that Theory Y is true. Which non-invasive measurement probes would you need now?
3. We discussed that we foresee two ways to collect measurements non- invasively: in batch and in background mode. What are the advantages and disadvantages of each approach?
4. There are many interrelated building blocks (or concepts) of Lean Software Development and each contributes differently to it. What types of data are handled by each of these building blocks (or concepts)? How do the contribution to the overall value stream, to the creation of knowledge, and to the overall improvement?
5. Assume you set up a fantastic dashboard for your team. As you collect the data and visualize it, you notice that all the measurements show problematic values. You let the dashboard in place for some days and also show it to your collaborators, but nobody cares; everybody continues his job as if everything would be fine. What is going wrong?

## K.42 Nonlinear Control Theory

- **Course name:** Nonlinear Control Theory
- **Course number:** XYZ
- **Knowledge area:** Control Engineering

### K.42.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 4th year of BS
- **Semester of instruction:** 2nd semester
- **No. of Credits:** 4 ECTS
- **Total workload on average:** 144 hours overall
- **Frontal lecture hours:** 2 per week
- **Frontal tutorial hours:** 2 per week
- **Lab hours:** 2 per week
- **Individual lab hours:** 0
- **Frequency:** weekly throughout the semester
- **Grading mode:** letters: A, B, C, D

### K.42.2 Prerequisites

- Physics I
- Physics II
- Mathematical Analysis I
- Mathematical Analysis II
- Analytic Geometry and Linear Algebra I
- Analytic Geometry and Linear Algebra II
- Control Theory

### K.42.3 Course outline

This introductory course considers the fundamental principles and techniques of nonlinear control systems, providing basic information on mathematical modeling, vector spaces and norms, Phase plane portraits, Lyapunov stability, Jacobian-based linearization and MIMO-based state-space representation. It covers stability analysis of nonlinear and time-varying systems, and Internal stability of feedback systems. During the course students will become familiar with control pronciples for nonlinear control systems. This course includes practical simulation exercises and a control design project.

### K.42.4 Expected learning outcomes

- Understanding nonlinear and time-varying systems
- Realizing the limits of nonlinear systems
- Amplitude, phase and frequency analysis of nonlinear control systems
- Techniques for the stability analysis of nonlinear and time-varying systems
- Jacobian linearization and gain scheduling
- State-Space Modelling

### K.42.5 Expected acquired core competences

- Time and Frequency analysis
- Building controllers
- Tools: MATLAB / Simulink
- Nonlinear control system analysis
- Control design for nonlinear systems

### K.42.6 Detailed topics covered in the course

The topics below are presented with the granularity of at most the academic hour of instruction. For each topic it is specified if it an **I**ntroduction to the topic, a **D**eep explanation, or a **R**eview of a subject already covered in another course.

- Introduction to nonlinear and time-varying systems
- Mathematical background, including vector spaces and norms
- Local Decompositions of Control Systems
- Global Decompositions of Control Systems

- Techniques for the stability analysis of nonlinear and time-varying systems
- Internal stability of feedback systems. Phase plane portraits
- Lyapunov stability theorems. Popov and circle criteria for nonlinear feedback systems
- Passivity and small gain for nonlinear operators
- Overview of design for nonlinear systems
- Jacobian linearization and gain scheduling
- Input-Output Maps and Realization Theory
- Theory of Nonlinear Feedback for Single-Input Single-Output Systems
- Theory of Nonlinear Feedback for Multi-Input Multi-Output Systems
- Geometric Theory of State Feedback
- Tracking and regulation
- Global feedback design for SISO Systems

### K.42.7 Expected learning outcomes

- Understanding in Nonlinear Control System characteristics: controllability, stability and regulation quality
- Realizing the limits of the Nonlinear Systems
- Amplitude, phase and frequency analysis of control systems
- Time analysis of control systems
- Building PID controllers and compensators
- State-Space Modelling
- State observer applying

### K.42.8 Expected acquired core competences

- Time and Frequency analysis for nonlinear control system
- Control system analysis
- Building controllers
- Tools: MATLAB / Simulink
- Nonlinear control system analysis

### K.42.9 Textbook

-

### K.42.10 Reference material

- Slides will be provided during the course

### K.42.11 Required computer resources

Students are required to have laptops.

### K.42.12 Evaluation

- Quizzes (20%)
- In-class activity (10%)
- Mid-term exam (5%)
- Final exam (10%)
- Home Assignments (30%)
- Project (25%)

## K.43 Robotic Systems

- **Course name:** Robotic Systems
- **Course number:**

### K.43.1 Course characteristics

### K.43.2 What subject area does your course (discipline) belong to?

Robotics; Robotic components; Robotic control.

### K.43.3 Key concepts of the class

- Robot kinematics and dynamics
- Manipulator compliance and stiffness
- Control approaches for robotic systems

### K.43.4 What is the purpose of this course?

The purpose of this course is to review approaches to mathematical modeling of robotic manipulators' kinematics and dynamics and then introduce the concepts of robot stiffness, identification of joint compliance, and basics of calibration. Based on these concepts, the course will outline general nonlinear approaches to control over robotic systems with compliance and teach the students to analyze stability of nonlinear control strategies.

### K.43.5 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember

- How to drive expressions for position kinematics and differential kinematics of serial manipulators,
- How to model dynamics of serial and parallel robots,
- What approaches exist to model robot joints' elasticity,

- Fundamental principles of position tracking control in manipulators,
- Energy-based and other nonlinear approaches to control over compliant robots.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- How to find Jacobian for series and parallel robots and use it to compute forces and torques,
- What constitutes a common manipulator calibration procedure,
- Reasons and examples of singularities for serial and parallel robots,
- What are limit cycles in robots with compliance,
- How to model and control tendon-driven robots.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Find stiffness matrix for given manipulator,
- Analyze joint constraints and find singularities,
- Design stable position controllers for manipulators,
- Develop nonlinear controllers such as gravity compensation and inverse dynamics.

### K.43.6 Course evaluation

**Table K.95:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 10 |
| Interim performance assessment | 30 | 60 |
| Exams | 50 | 30 |

If necessary, please indicate freely your course's features in terms of students' performance assessment:

The course grades are given according to the following rules: In-class discussion and lab performance = 10 pts, Homework assignments (4) = 20 pts, Quizzes (4) = 20 pts, Exams = 30 pts, Term project = 20 pts.

### K.43.7 Grades range

**Table K.96:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

If necessary, please indicate freely your course's grading features.

### K.43.8 Resources and reference material

Textbooks:

-
-
-
-
-
-

### K.43.9 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Robot kinematics and dynamics | 4 |
| 2 | Manipulator stiffness and calibration | 8 |
| 3 | Position tracking in robots | 6 |
| 4 | Nonlinear control approaches | 6 |

#### K.43.9.1 Section 1

**Section title:** Robot kinematics and dynamics

**Topics covered in this section:**

- Position and velocity kinematics
- Dynamics of rigid manipulators
- Virtual joint modeling
- Matrix structural analysis

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

- Given initial and final object position and orientations, obtain the corresponding transformation matrix.
- Analyze given Jacobian to find kinematic singularities.
- For a given manipulator with known Jacobian, do the following:
  - Find kinetic and potential energies of the robot;
  - Drive the Euler-Lagrange differential equation of motion.
- Compare solving forward and inverse kinematics problems for serial and parallel manipulators.

**Typical questions for seminar classes (labs) within this section**

1. Solve forward and inverse kinematics problems for a given manipulator.
2. Given the kinematic model of the FANUC R-2000iC/165F robot, find its Jacobian and analyze kinematic singularities.
3. Write differential equations of a given manipulator using Euler-Lagrange formulation.

4. Write forward dynamics for a robot using Newton-Euler method.

**Test questions for final assessment in this section**

1. What is manipulator Jacobian? How does it relate static forces and torques? How can one use the Jacobian to analyze manipulator singularities?
2. What are the properties of the main terms of differential equations of manipulator motion (invertibility, positive definiteness, singularities, limits).
3. What are the principal differences in kinematics and dynamics of serial and parallel robots?

#### K.43.9.2 Section 2

**Section title:** Robot calibration

**Topics covered in this section:**

- Joint stiffness and compliance
- Virtual joint modeling
- Matrix structural analysis
- Sources of uncertainties and model errors in practical robots
- Types of robot calibration
- Complete, irreducible geometric models
- Elastostatic calibration

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is stiffness matrix of manipulator and what does it describe?
2. Why is robot calibration needed?

3. What are the main sources of errors in robot parameters?
4. Give examples of geometric and non-geometric errors.
5. Describe typical steps of calibration procedure.

**Typical questions for seminar classes (labs) within this section**

1. Find stiffness matrix of a given parallel robotic platform.
2. Perform matrix structural analysis of a cantilever beam.
3. Find stiffness matrix of a two-link manipulator with elastic joint.
4. Estimate identification accuracy for 3-link manipulator.
5. Comment on differences between compliance matrix of a manipulator obtained via CAD modeling and identification results.

**Test questions for final assessment in this section**

1. Describe main stiffness modeling approaches, their particularities, advantages and limitations.
2. Use variable joint model for a serial manipulator (assume all elements are flexible) to find stiffness matrix.
3. Describe particularities and difficulties of the elastostatic calibration.
4. What is complete, irreducible geometric model and why do we need it?

#### K.43.9.3 Section 3

**Section title:** Position tracking in robots

**Topics covered in this section:**

- Linear feedback control in application to manipulators
- Lyapunov stability analysis
- Stabilization and trajectory tracking

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Give an example of using feedback in activities of daily life.
2. Drive error dynamics equations for a given feedback control law.

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

3. For a given differential equation describing robot dynamics and several control laws, find which ones are stable using Lyapunov stability theory.
4. How does elasticity affect error dynamics in robot control applications?
5. What components of mechanical energy exist in robots with compliance?

**Typical questions for seminar classes (labs) within this section**

1. Design PD controller for a rigid two-link manipulator.
2. Numerically model behavior of compliant robot with PID controller.
3. Compare performance of linear controllers in application to rigid and 2-link manipulator control.
4. Analyze stability of a given controller.

**Test questions for final assessment in this section**

1. What challenges does robot compliance pose for a control system?
2. Design a position tracking controller for a given compliant system.
3. How does cable elasticity affect dynamics and control of tendon-driven robots?

#### K.43.9.4 Section 4

**Section title:** Energy, impedance, and force control

**Topics covered in this section:**

- Joint-space inverse dynamics of serial manipulators
- Energy-based control of compliant robots
- Passivity-based control
- Adaptive control of flexible joint manipulators

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Provide examples of passive and active systems.
2. What are limit cycles?
3. What happens with the energy of passive systems with time?
4. For a given differential equation that describes pendulum dynamics, do the following:
5. Perform input-state linearization for a given system of differential equations.

**Typical questions for seminar classes (labs) within this section**

1. Find limit cycles of a given robot with compliance.
2. Design gravity and compliance compensator for a robot with flexible joints.
3. Implement and simulate passivity-based control over given robot.
4. For a given differential equation that describes pendulum dynamics, do the following:
   - Find control law transforming original dynamics into that of a linear mass-spring-damper system;
   - Write position error dynamics for the designed control law (inverse dynamics);
   - Repeat the previous steps if there are uncertainties in some of the system's parameters.

**Test questions for final assessment in this section**

1. Provide examples of practical systems with non-collocated feedback. What unique challenges does this pose for control systems?
2. Design a position tracking controller for a given compliant system.
3. Analyze stability of a given system with passivity-based controller

4. What are the physical fundamentals behind the concept of passivity and passivity-based control?

## K.44 Sensors and Sensing

- **Course name:** Sensors and Sensing
- **Course number:**

### K.44.1 Course Characteristics

### K.44.2 What subject area does your course (discipline) belong to?

Hardware; Sensors and actuators; Robotic components.

#### K.44.2.1 Key concepts of the class

- Physical principles for sensors and sensing
- Basics of error analysis and calibration
- Sensor's data processing and filtering

#### K.44.2.2 What is the purpose of this course?

This course covers selected topics in sensors and sensing area, which are in particular important for robotic application. The students are expected to learn the course topics on their own beyond the level of the lectures. The goals throughout the course are to refresh students' math skills in data and error analysis, and familiarize them with sensing principles and sensor utilization, giving them both analytic and experimental experience. The students will be required to participate in laboratory practicum and solve practical tasks on data processing in MATLAB/Python/C++ environment.

#### K.44.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize

- Various types of sensors, their pros and cons,
- Working principles of electric actuators (DC motors),
- Operation principles of motion transmission mechanisms,

- Fundamentals of linear feedback control systems, and
- Principles of controller design for mechatronic systems.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- How to select sensors for a given application,
- How to choose appropriate transmission mechanisms and account for their efficiency,
- How to integrate all selected parts to create a mechatronic system,
- Typical nonlinearities that originate from electronic and mechanical sources and their effects on system performance, and
- How to tune control system for selected motor and desired performance specifications.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Drive differential equations of motion describing behavior of physical systems with several degrees of freedom,
- Calculate motor and sensor requirements for a given physical system, control task or application,
- Select appropriate motor that provides enough power while avoiding overheating,
- Tune control system for selected motor, transmission mechanism and sensor to achieve desired response and stability.

### K.44.3 Expected learning outcomes

- Be acquainted with and utilize main sensors for robotic applications.
- Be familiar with Sensing principles, Measurements and error analysis, Data analysis, Sensor calibration, etc.
- Be able to write a code for sensor data processing in MATLAB.

- Train engineering skills during lab. tests, studying sensor calibration and utilization, and using relevant software for sensor data capture, export, interpretation and representation.

### K.44.4 Expected acquired core competences

- Know sensors physical principles and their limitations.
- Be capable to describe sensors work and regions for applications.
- Be familiar with Sensor's calibration and exploitation.
- Be able to introduce and perform Measurements, Data and Error analysis.
- Be acquainted with and utilize Linear regression, Least-squares fitting, Curve fitting, and Filtering procedures.
- Demonstrate ability to process sensor data by relevant math. methods.
- Write MATLAB code for sensor data processing.

### K.44.5 Detailed topics covered in the course

The topics below are presented with the granularity of at most the academic hour of instruction. For each topic it is specified if it an **I**ntroduction to the topic, a **D**eep explanation, or a **R**eview of a subject already covered in another course.

- Introduction to sensors: Sensors' classification, Characteristics, Dynamic range Accuracy
- Introduction to Measurements and Error Analysis
- Introduction to Data Analysis: Linear Regression, Least-Squares Fitting, Curve fitting, and Filtering
- Image sensors: camera matrix, characteristics and calibration
- Video camera: CCTV, IR & thermal imaging camera, Fish eye camera
- Stereo vision: Stereosystem, Stereogeometry, 3D reconstruction
- Depth, TOF, RGBD camera, MS Kinect: characteristics and calibration
- Review and Midterm
- Sensor fusion (principles), Multisensory, multicamera, MoCap systems
- LIDAR: Laser rangefinders. Laser-camera systems
- SONAR. Doppler radar. Acoustic sensor systems. Sound spectrogram

- Inertial sensors: IMU, accelerometers, gyroscopes, Magnetic Compasses, GPS
- Internal sensors: position, velocity, torque & force sensors, encoders
- MEMS for robot applications. Smart and Intelligent Sensors

### K.44.6 Textbook

-

### K.44.7 Reference material

- Slides will be provided during the course

### K.44.8 Required computer resources

Students are required to have laptops.

### K.44.9 Evaluation

- Home assignments (25%)
- Mid-term exam (15%)
- In-class activity, quizzes and lab. practicum (15%)
- Final exam (25%)
- Project (20%)

## K.45 Mechanics and Machines

- **Course name:** Mechanics and Machines
- **Course number:**

### K.45.1 Course Characteristics

### K.45.2 What subject area does your course (discipline) belong to?

Mechanical engineering, modeling and design of mechanisms and machines

#### K.45.2.1 Key concepts of the class

- Mechanical engineering: Methods of calculation and design of mechanisms and machines

#### K.45.2.2 What is the purpose of this course?

The development of any classes of robots and the use of robots in industry requires the engineer to have knowledge and skills in the analysis and synthesis of mechanisms, the dynamic calculation of mechanisms and machines, the calculation of strength and rigidity, the ability to read drawings and work in modern CAD and CAE systems.

The purpose of the course is to give broad basic knowledge in mechanical engineering and to show the modern capabilities of computer technology for solving engineering problems. The course covers topics such as the strength of materials and the theory of mechanisms and machines. The objective of the course is to provide knowledge and skills that are useful in the development of new robots, and are also necessary for the effective use of industrial robots for various types of material processing.

#### K.45.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize

- Various types of joining machine parts,

- Various methods of manufacturing machine parts and processing their surfaces,
- Various types of mechanisms and their scope,
- Various types of gears and their main characteristics,
- The main problems of the structural, kinematic and dynamic analysis of mechanisms,
- Fundamental principles of strength of materials,
- The main problems of strength and stiffness calculation.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- How to read engineering drawings,
- How to choose a method of joining machine parts,
- How to choose the type of gears,
- How to conduct structural, kinematic and dynamic analysis of mechanisms,
- How to balance mechanisms,
- How to choose the shape of the machine part depending on the type of loading,
- How to evaluate the strength and stiffness of machine parts

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Design simple machinery using a CAD system
- Perform kinematic and dynamic calculations of mechanisms and machines
- Synthesize and optimize mechanisms in accordance with specified requirements
- Perform strength and stiffness calculations manually and using a CAD system

#### K.45.2.4 Course evaluation

The course grades are given according to the following rules: Research project = 50 pts, Quizzes (5) = 10 pts, Midterm exam = 20 pts, Final exam 20 pts.

**Table K.97:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 10 |
| Interim performance assessment | 30 | 50 |
| Exams | 50 | 40 |

**Table K.98:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

#### K.45.2.5 Grades range

#### K.45.2.6 Resources and reference material

Main textbook:

- Michael M. Stanišic Mechanisms and Machines: Kinematics, Dynamics, and Synthesis, 2015
- Dietmar Gross et al. Engineering Mechanics 2: Mechanics of Materials, 2018

Other reference material:

- Vitor Dias da Silva Mechanics and Strength of Materials 2006
- Frolov K.V. at al. Teoriya mechanizmov i mashin 1987 - in Russian
- Artobolevski I.I. Teoriya mechanizmov i mashin 1988 - in Russian
- Artobolevski I.I., Edelstein B.V. Sbornik zadach po teorii mechanizmov i mashin 1975 - in Russian
- Jukov V.G. Mechanika. Soprotivlenie materialov 2012 - in Russian
- Volmir A.S. at al. Sbornik zadach po soprotivleniu materialov 1984 - in Russian
- Mary Kathryn Thompson, John Martin Thompson ANSYS Mechanical APDL for Finite Element Analysis, 2017
- Alawadhi, Esam M. Finite element simulations using ANSYS, 2016
- Fedorova N.N. at al. Osnovy paboty v ANSYS 17, 2017 - in Russian

### K.45.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.99:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction to Engineering | 10 |
| 2 | Theory of mechanisms and machines | 10 |
| 3 | Strength of materials | 10 |

#### K.45.3.1 Section 1

**Section title:** Introduction to Engineering

**Topics covered in this section:**

- Reading engineering drawings.
- Types of joining machine parts.
- Overview of axles, shafts, bearings, couplings.
- Overview of materials used in mechanical engineering.
- Overview of parts manufacturing methods.
- Overview of surface treatments.
- Application of CAD / CAE systems for the design of machine parts.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Make projections the part according to its 3D model or image.
2. Make a 3D model of the part according to its drawing.

3. Suggest a method of manufacturing a part according to its drawing and description.
4. Suggest a way to connect parts according to their description.
5. Suggest a method for transmitting movement in a machinery by describing its functions.

**Typical questions for seminar classes (labs) within this section**

1. Analyze the parts of a given mechanism or construction.
2. Using a CAD-system, make solid-state 3D-models of parts of a given mechanism or construction
3. Make a 3D model of the assembly of a given mechanism or construction
4. Make and describe drawings of parts of a given mechanism or construction

**Test questions for final assessment in this section**

1. Describe the principles of solid modeling in CAD systems.
2. Describe the scope, advantages and disadvantages of various ways of joining machine parts.
3. Describe the scope, advantages and disadvantages of various bearings.
4. Describe the basic methods for manufacturing parts and processing their surfaces, their advantages and disadvantages.

#### K.45.3.2 Section 2

**Section title:** Theory of mechanisms and machines

**Topics covered in this section:**

- Structural analysis of mechanisms
- Kinematic analysis of mechanisms
- Kinematic analysis of gears
- Force and dynamic analysis of mechanisms
- Dynamic characteristics of machines
- Synthesis of mechanisms
- Balancing mechanisms

- Vibration and vibration protection
- Application of CAD / CAE systems for kinematic and dynamic analysis of mechanisms

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Propose the structure of the mechanism for the implementation of the given functions
2. Calculate the trajectories, velocities and accelerations of specific points and links of the mechanism
3. Calculate gear transfer functions
4. Calculate the forces arising in the mechanism during its movement
5. Calculate mechanism balancing parameters
6. Make a 3D model of the mechanism in a CAD system, check its operability

**Typical questions for seminar classes (labs) within this section**

1. Apply kinematic and dynamic analysis of mechanisms and gears.
2. Perform a synthesis of mechanisms to implement the specified functions.
3. Make engine selection according to dynamic characteristics.
4. Calculate balancing mechanisms, including the use of flywheels.
5. Perform vibration analysis and vibration protection design.

**Test questions for final assessment in this section**

1. Describe the different types of gears, their advantages and disadvantages.
2. Describe the method of selecting the type of gears.
3. Describe the main problems of structural, kinematic and dynamic analysis of mechanisms.

4. Describe the methods of kinematic and dynamic analysis on the example of a given mechanism.
5. Describe the methods of balancing mechanisms

#### K.45.3.3 Section 3

**Section title:** Strength of materials

**Topics covered in this section:**

- Introduction to the resistance of materials
- Stresses and deformations
- Stretching and compression
- Pure shift
- Torsion
- Bending
- Dynamic loads
- Cyclic loads

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Calculate stresses and strains in a bar loaded with stretching or compressive force.
2. Calculate stresses in shear stresses in riveted or welded joints.
3. Calculate stress and strain of a shaft loaded with torque.
4. Calculate stress and strain of a beam loaded with bending moments.
5. Using a CAD system, calculate stresses and deformations of parts of the developed mechanism.

**Typical questions for seminar classes (labs) within this section**

1. Apply methods of resistance of materials to assess the operability of parts in cases of simple loading.
2. Apply a CAD system to calculate and evaluate the operability of parts in cases of complex loading.
3. Design and optimize the shape of parts depending on the conditions of their loading.
4. Apply a method for evaluating the durability and reliability of parts.

**Test questions for final assessment in this section**

1. Describe the fundamental principles and basic laws of strength of materials.
2. Describe the calculation method for strength and stiffness in cases of stretching/compression, shift, torsion and bending.
3. Describe the method of selecting the shape of the machine part, depending on the type of loading.
4. Describe methods for evaluating the durability and reliability of machine parts.

## K.46 Network and CyberSecurity

- **Course name:** Network and Cyber Security
- **Course number:** ?

### K.46.1 Course characteristics

### K.46.2 Key concepts of the class

- Information Security Management
- Web Security
- Software Security
- Network security

### K.46.3 What is the purpose of this course?

Security breaches cost billions of dollars worth of damage to the computing industry. Today, cybercriminals control armies consisting of several millions of compromised machines. Attacks are increasingly being perpetrated towards enterprises, individuals, critical infrastructure and even governments. At the same time, our computer systems and platforms are fast evolving to meet the demands of the industry. Increasing the use of personalized devices, and our growing dependence on legacy computer systems that weren't designed with security in mind is a challenge ahead. Therefore, the purpose of this course is to cover the design and implementation of different IT systems from a security perspective. This course introduces to the field of systems security: that is, how to analyze and develop secure systems. The course covers fundamental concepts of systems design, low and high-level vulnerabilities exploitation, design, and implementation flaws in different types of applications based on the real-world scenarios.

### K.46.4 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define

- Security policies and controls
- Risks and threats related to the system design and its implementation
- Software security testing methodologies
- Software development security techniques
- Injection and authorization flaws
- Cookies and misconfiguration flaws
- Common weaknesses/vulnerabilities in web applications
- Common weaknesses/vulnerabilities in the typical systems software

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- Information security management methods
- Difference between different types of risks and threats
- Security-related web technologies
- The difference in the different web application flaws
- ASLR, NX and how are these techniques can help to protect against a malicious attacker
- Covert channels
- Networking tools
- Network proxies

**- What should a student be able to perform at the end of the course?**

- Critically audit systems and code for security flaws and threats
- Design and implement exploits for real security vulnerabilities
- Develop secure systems and applications
- Be able to design defense solutions and outline their limitations
- Be able to find misconfigurations/vulnerabilities in a given network/system

**Table K.100:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 30 |
| Project | 30 | 30 |
| Exams | 50 | 40 |

### K.46.5 Course evaluation

If necessary, please indicate freely your course's features in terms of students' performance assessment:

### K.46.6 Grades range

**Table K.101:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 90-100 |
| B. Good | 75-89 | 70-89 |
| C. Satisfactory | 60-74 | 60-69 |
| D. Poor | 0-59 | 0-59 |

If necessary, please indicate freely your course's grading features:

The laboratory assignments are mandatory with a requried minimum result of 6/10 - including re-takes and late submissions - to complete the course. As a conscequence, the grades are generally pretty high and therefore the grading ranges are scaled up.

### K.46.7 Resources and reference material

- Mike Chapple, James Michael Stewart, Darril Gibson, CISSP Official Study Guide, 8th Edition, Sybex, 2018
- Michal Zalewsk, The Tangled Web, No Starch Press, 2011
- Jon Erickson, Hacking: The Art of Exploitation, 2nd Edition, No Starch Press, 2008

**Table K.102:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Information security management | 6 |
| 2 | Web security | 6 |
| 3 | Software Security | 8 |
| 4 | Network Security | 4 |
| 5 | Labs | 56 |

## K.46.8 Course Sections

## K.46.9 Section 1

**Section title:** Information Security Management

**Topics covered in this section:**

- Security Policies and Controls
- Risks Analysis and Threats Modeling
- Software Development Security Techniques

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

- What types of Security Policies are exist?
- What information from a given system you need to take into account to calculate security risks?

- Explain the difference between static and dynamic analysis of application code?

**Typical questions for seminar classes (labs) within this section**

- Audit the given security policy for vulnerabilities and update it accordingly
- Calculate security risks for a given system and develop necessary security measures for mitigation

**Test questions for final assessment in this section**

As above

### K.46.10 Section 2

**Section title:** Web Security

**Topics covered in this section:**

- Security-related web technologies
- Same Origin Policy
- Web Attacker Model
- OWASP methodology
- Injection Flaws
- Authorization Flaws
- Cookies Flaws
- Server Misconfiguration

**What forms of evaluation were used to test students' performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

- What is the difference between reflected XSS and stored XSS? which one is more critical and why?
- What are the pros and cons of using regex to protect against XSS?
- what is the Same Origin Policy? and which attack does it mitigate?
- What is the difference between boolean-based and time-based SQL injection?

**Typical questions for seminar classes (labs) within this section**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

- Vulnerability analysis and exploitation for a given web application
- Write and deploy WAF rules to mitigate a specific web attack
- Does the Same Origin Policy apply to the localStorage inside the browser?

**Test questions for final assessment in this section**

As above

### K.46.11 Section 3

**Section title:** Software Security

**Topics covered in this section:**

- Binary Exploitation
- ASLR
- NX

**What forms of evaluation were used to test students' performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

- What are the pros and cons of using ASLR? does it affect the performance?
- What can you do with a format string vulnerability?
- What is the required information to be able to identify a remote libc version?
- Why some binaries might have the same address for their functions? what is the security risk of this?

**Typical questions for seminar classes (labs) within this section**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

- Vulnerability analysis and exploitation for a given binary while ASLR is disabled
- Try to rewrite the following Assembly code in any programming language
- How can you check if you have ASLR, PIE, NX enabled or not?
- Decompilers are not always accurate why? how can you improve it?
- Some binaries are shipped with debugging symbols, How can this help you in debugging?

**Test questions for final assessment in this section**

As above

### K.46.12 Section 4

**Section title:** Network Security

**Topics covered in this section:**

- Networking tools
- Network attacks
- IDS/IPS

**What forms of evaluation were used to test students' performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

- What is the difference between VPN and sock5?
- What are IDS, IPS, and DPI?
- Why does Nmap produce false-positive when scanning a windows host? can you improve the scanning technique?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

- What is covert channel? what are the most common protocols that are used for covert channel? why?
- When using a proxy for HTTPS, your browser will always complain about the certificate, how can you solve this issue?

**Typical questions for seminar classes (labs) within this section**

- No lab for this section

**Test questions for final assessment in this section**

As above

## K.47 Statistical Techniques for Data Science

- **Course name:** Statistical Techniques for Data Science
- **Course number:** BS-STDS

### K.47.1 Course characteristics

#### K.47.1.1 Key concepts of the class

- Statistical Hypothesis Testing
- Resampling
- Statistical ML
- MCMC

#### K.47.1.2 What is the purpose of this course?

The course covers non-standard statistics, applicable in a wide set of contexts, including non-parametric statistics, simulation methods, and time series analysis.

This course will provide an opportunity for participants to learn: random variables, elementary probability, and distributions; relevant probabilistic inequalities; random vectors, marginal and joint distributions; sequences of random variables and concepts of convergences; Markov chains; processes in continuous time; univariate and multivariate simulation methods; non-parametric and parametric resampling methods.

#### K.47.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define

- Estimation methods: point estimates, MLE
- Confidence interval, p-value
- Estimation and Non-parametric Tests. Kolmogorov-Smirnov Test
- Sampling. Metropolis-Hastings. Markov Chains. MCMC

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- Describe the Statistical Hypothesis Testing, p-value, Power of a test and Sample size
- Explain ANOVA, Chi-square tests
- Smoothing methods with examples

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply

- Apply Non-parametric Tests, such as KS-test
- Apply resampling methods (jackknife, bootstrap)
- Apply Markov chain Monte-Carlo methods

#### K.47.1.4 Course evaluation

**Table K.103:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 40 |
| Interim performance assessment | 30 | 30 |
| Exams | 50 | 30 |

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

#### K.47.1.5 Grades range

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table M.73, but it may change slightly (usually reduced) depending on how the semester progresses.

**Table K.104:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 65-79 |
| C. Satisfactory | 60-74 | 50-64 |
| D. Poor | 0-59 | 0-49 |

#### K.47.1.6 Resources and reference material

- Murphy K.P. Machine Learning: A Probabilistic Perspective. Massachusetts Institute of Technology, 2012. — 1067 p.
- Bishop Christopher. Pattern Recognition and Machine Learning. Springer, 2006. — 738 p.
- M. Ross. Introduction to Statistics. Prentice Hall. 1989
- Efron, R. J. Tibshirani. An introduction to the bootstrap. Springer. 1993
- G. Casella, R. L. Berger. Statistical Inference. Thomson Press. 2006
- S. Hojsgaard, D. Edwards, S. Lauritzen. Graphical Models with R. Springer. 2012
- Hastie, T. Tibshirani, R. and Friedman, J. (2008) The Elements of Statistical Learning 2ed. Springer
- Steven M. Kay. Fundamentals of Statistical Signal Processing: Estimation Theory (v. 1). Prentice Hall. 1993

### K.47.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.105:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Parametric Statistics | 42 |
| 2 | Non-parametric Statistics | 24 |
| 3 | Sampling and Simulation | 24 |

#### K.47.2.1 Section 1

**Section title:**

Parametric Statistics

**Topics covered in this section:**

- Review of Probability Theory. Random variables. Density. Distributions. Expected value
- Exploring the Data Distributions. Multivariate distributions. Plots
- Data and Sampling Distributions. Standard error. CLT
- Experiment Design. Confidence intervals. Introduction to Hypotheses Testing
- A/B Testing, T-test, ANOVA, Chi-square.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the Central Limit Theorem?
2. What is statistic?
3. What is sampling distribution?
4. What is standard error?
5. What are Type I and Type II errors?
6. What is t-statistic error? T-test?

**Typical questions for seminar classes (labs) within this section**

1. Create a bi-modal dataset, which has the mean less than the median, draw a histogram.
2. Poisson Distribution in practice: seatching patterns of palindromes in DNA.
3. Experiments and A/B testing.

4. A researcher claims that Democrats will win the next election. 4300 voters were polled; 2200 said they would vote Democrat. Decide if you should support or reject null hypothesis. Is there enough evidence at *alpha* = 0.05 to support this claim?

**Test questions for final assessment in this section**

1. Prove Chebyshov inequality
2. Prove Markov inequality
3. What is ANOVA, what is the difference with Chi-square test?

### K.47.2.2 Section 2

**Section title:**

Non-parametric Statistics

**Topics covered in this section:**

- Empirical CDF. Resampling. Jackknife and Bootstrap
- Density Estimation
- Estimation and Non-parametric Tests. KS Test
- Non-parametric Tests. Kruskal-Wallis Test. Multi-arm Bandits

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is Empirical CDF?
2. How to apply resampling? Jackknife and Bootstrap?
3. What is Kernel Density Estimation?
4. What is Smoothing?

**Typical questions for seminar classes (labs) within this section**

1. Implement Kernel Density Estimation.
2. Apply KS-test.
3. Apply Kruskal-Wallis Test
4. Implement Multi-arm Bandits

**Test questions for final assessment in this section**

1. What is epsilon-greedy algorithm?
2. Perform 1 sample KS test in Python and Scipy. Compare KS test to visual approaches for checking normality assumptions
3. Plot CDF and ECDF to visualize parametric and empirical cumulative distribution functions

#### K.47.2.3 Section 3

**Section title:**

Sampling and Simulation

**Topics covered in this section:**

- Sampling. Metropolis-Hastings.
- Rejection Sampling. Gibbs Sampling
- Thompson Sampling. Upper confidence bound
- Markov Chains. MCMC
- Time Series: Tools and Applications

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is Thompson Sampling?
2. What Upper confidence bound algorithm?
3. What is stationary distribution?

**Typical questions for seminar classes (labs) within this section**

1. Given density function, implement Accept-Reject sampling
2. Run Metropolis Hastings and Accept-Reject (on the same f(x)) (n=1000, 10000, 100000). Compare results
3. Apply Gibbs Sampling
4. Apply tools for time series analysis and prediction

**Test questions for final assessment in this section**

1. Consider a transition matrix of a Markov Chain (MC).
   - Show that this is a regular MC (A MC is called regular if for some integer n all entries of transition matrix after n steps are strictly positive).
   - Find the limiting probability vector w
2. Compare Gibbs Sampling to Metropolis Hastings.

# K.48 Numerical Modelling

- **Course name:** Numerical Modelling
- **Course number:** XYZ
- **Subject area:** Math

## K.48.1 Course characteristics

### K.48.1.1 Key concepts of the class

- An understanding of the basic "canon" of numerical algorithms and numerical methods relevant to computing given task

### K.48.1.2 What is the purpose of this course?

This course answer on the next questions. To what problems does an algorithm or method apply? How does the method work? How does the method compare to alternatives (in terms of appropriate computational metrics)? What can go wrong? What are the sources of error and uncertainty?

### K.48.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

- Understand key principles involved in numerical solution of typical mathematical problems.
- Become familiar with numerical differentiation and integration.
- Solve systems of non/linear algebraic equations numerically by different ways.
- Become familiar with methods of interpolation and regression.
- Get hands-on experience with numerical solving system of nonlinear differential equations.

**- What should a student be able to understand at the end of the course?**

- Key principles involved in numerical solution of typical mathematical problems.
- How to apply numerical differentiation and integration.

- How to solve systems of non/linear algebraic equations numerically by different ways.
- How to apply methods of interpolation and regression.
- How to make numerical solving system of nonlinear differential equations.

**- What should a student be able to apply at the end of the course?**

- Numerical solution of typical mathematical problems
- Make nonlinear regression and interpolation
- Make numerical differentiation and integration
- Numerical solution systems of non/linear algebraic equations
- Numerical solving system of nonlinear differential equations

#### K.48.1.4 Course evaluation

**Table K.106:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 30 |
| Interim performance assessment | 30 | 30 |
| Exams | 50 | 50 |

#### K.48.1.5 Grades range

**Table K.107:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 85-100 |
| B. Good | 75-89 | 65-84 |
| C. Satisfactory | 60-74 | 50-64 |
| D. Poor | 0-59 | 0-49 |

#### K.48.1.6 Resources and reference material

**Textbooks:**

- Gilbert Strang. Computational Science and Engineering. Wellesley, MA: Wellesley-Cambridge Press, 2007. 727 Pg. ISBN: 9780961408817.
- I.B. Petrov, A.I. Lobanov. Lectures in Computational Mathematics. M.: Internet University of Information Technology, 2006. 523 c. ISBN: 5-94774-542-9.

**Reference material:**

- Jaan Kiusalaas. Numerical Methods in Engineering with Python. Cambridge University Press, 2005. 433 Pg. ISBN: 978-0-521-85287-6.

## K.48.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.108:** Course Sections

| Section Number | Section Title | Lectures (hours) | Seminars (labs) | Self-study | Knowledge evaluation |
|---|---|---|---|---|---|
| 1 | Numerical differentiation and integration, functions interpolation, solution of system of linear algebraic equations. | 14 | 14 | 28 | 2 |
| 2 | Solution of nonlinear algebraic equations and systems. Solving of ODEs and PDEs. Discrete Fourier Series. | 12 | 12 | 24 | 2 |
| Final examination | | | | | 2 |

### K.48.2.1 Section 1

**Section title:** Numerical differentiation and integration, functions interpolation, solution of system of linear algebraic equations

**Topics covered in this section:**

- Key concerns of numerical computations. Accuracy of floating-point arithmetic.
- Numerical differentiation. Method of undetermined coefficients.
- Interpolation of functions. Splines.
- Numerical integration. Quadrature formulas.
- Solution of system of linear algebraic equations.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How to perform numerical differentiation by the method of undetermined coefficients?
2. How to perform interpolation of function by using splines?
3. How to perform numerical integration by using quadrature formulas?
4. How to solve a system of linear algebraic equations by using iteration methods?
5. How to solve a system of linear algebraic equations by using variation methods?

**Typical questions for seminar classes (labs) within this section**

1. To make numerical differentiation by the method of undetermined coefficients.
2. To make function interpolation by using splines.
3. To make numerical integration by using quadrature formulas.
4. To Solve a system of linear algebraic equations by using iteration methods?
5. To solve a system of linear algebraic equations by using variation methods?

**Test tasks for final assessment in this section**

1. You are requested to compute the integral of a black box function. The function would be supplied to you at the compile time in the form of a header file $blackbox.h$. At the very beginning of your program you should read single integer $n$ from the standard input call the function $blackbox - init(n)$. $blackbox - init$ should only be called once. All other functions should only be called after $blackbox - init$. Calling $blackbox_i nit$ with argument different from the one supplied via standard input leads to undefined behaviour.

When you want to get the value of the function at point $x$ you should call $blackbox(x)$. This function is guaranteed to be thread-safe. $x$ should be in range [-1; 1].
If you want to get the maximum absolute value of the $k-th$ derivative of the black box function on the integration interval, you should call $blackbox-df(k)$. The $k$ should be integer from 1 to 6.
To check if the black box function is oscillating you should call $blackbox-period()$.
The returned value would be the period length if the function is oscillating, and 0 otherwise.
The required absolute precision is $10^-9$. The truncated $blackbox.h$ file (implementing only one of possible blackbox functions) and an example (suboptimal) solution $solution.cpp$ are available to you on the “Files” tab in PCMS.
You should only submit your $solution.c/solution.cpp$ file. The appropriate $blackbox.h$ will be supplied by the testing system.
You should not try to reverse-engineer the black box and/or interact with it in any way except through the four functions listed above.

2. The task is simple: you have to fit a set of points with a 9-degree polynomial $y = a_9x^9 + a_8x^8 + ... + a_1x + a_0$.
Your program receives the following stream of commands:
   - ADD $x$ $y$ Read values $x$ and $y$.
   - FIT Print the coefficients for a polynomial fitted through all the points read since the beginning of the program. You will have no more than 13 FIT commands in each test.
   - END Print the coefficients for a polynomial fitted through all the points read since the beginning of the program, and quit.

   You will get no more than 107 commands before END.
3. The task is simple: you have to solve a system of linear algebraic equations (SLAE) $Ax = b$ with residual no more than $10^-9$.
And the matrix $A$ is very nice: non-singular, symmetric and strictly diagonally dominant. A piece of cake, you might think. The catch: you do not have $A$ in any explicit form.

You can only get the result of its multiplication with the vector.
You have a number of the blackbox functions through which you work with the SLAE:

- void $blackbox - init()$ – initializes the internal blackbox data structures. Should be called in the very beginning of the program! No other blackbox function should be called before it, and no reading from the $stdin$ shall be made (or at least, as they say, "be kind, rewind").
- int $blackbox - size()$ – returns the number of equations (which is equal to the number of unknowns) of the system. The number of equations lies between 10 and 10000 (inclusive).
- void $blackbox - mult$ (const double $*x$, double $*out$) – compute the product of $A$ and vector $x$, write the results to out. The pointers $x$ and out should point to different chunks of memory of size at least $blackbox - size() * sizeof(double)$ bytes each.
- void $blackbox - rhs(double * b)$ – write the right-hand side of the SLAE (i.e., vector $b$) to the array $b$. The pointer $b$ should point to the chunk if memory of size at least $blackbox - size() * sizeof(double)$ bytes.
- void $blackbox - submit(double * solution)$ – write the result of the program. The array solution should contain the solution to the SLAE: $blackbox - size()$ values of type double. This should the last function to be called by your program (besides return 0;).

#### K.48.2.2 Section 2

**Section title:** Solution of nonlinear algebraic equations and systems. Solving of ODEs and PDEs. Discrete Fourier Series.

**Topics covered in this section:**

- Numerical solution of nonlinear algebraic equations and systems.
- Basic concepts of the theory of difference schemes. Numerical methods for solving of initial value problem for ordinary differential equations (ODEs).
- Numerical methods for solving of boundary value problems for ODEs.

- Discrete Fourier Series. Numerical solution of second-order ODEs by Discrete Fourier Series. Numerical solution of the partial differential equations (PDEs) by Discrete Fourier Series.
- Variable-directions method. Numerical solution of the PDEs by finite difference methods.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How to perform numerical solution of nonlinear algebraic equations and systems.
2. How to perform numerical solution of initial value problem for ordinary differential equations (ODEs).
3. How to perform numerical solution of boundary value problems for ODEs.
4. How to perform numerical solution of ODEs and PDEs by Discrete Fourier Series.
5. How to perform numerical solution of the PDEs by finite difference methods.

**Typical questions for seminar classes (labs) within this section**

1. To make numerical solution of nonlinear algebraic equations and systems.
2. To make numerical solution of initial value problem for ordinary differential equations (ODEs).
3. To make numerical solution of boundary value problems for ODEs.
4. To make numerical solution of ODEs and PDEs by Discrete Fourier Series.
5. To make numerical solution of the PDEs by finite difference methods.

**Test tasks for final assessment in this section**

1. You have to build software for a new GPS/GLONASS receiver. The satellite navigation works as follows (of course, this is a rather simplified description of a real-world situation). There are $N < 30$ satellites. Every satellite broadcasts its position $(x_i; y_i; z_i)$ and high-precision synchronized time $t_i$. These signals take time to reach the receiver (e.g., the one that's in your smartphone). If the receiver has position $x$; $y$; $z$, and receives the signal at time $t$, the following equation (called "Navigation equation") holds true (For simplicity, we choose the speed of light $c = 1$):
   $(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = (t - t_i)^2$
   As we can see, we have four unknowns (the position of the receiver $x$; $y$; $z$ and the precise time $t$ when it received the signal). So, we need at least $N = 4$ satellites to get the location of the receiver ("fix"). The system of exactly four navigational equations might, in general, have multiple solutions. But usually, more than $N > 4$ satellites are visible, and we have an overdetermined system of nonlinear equations (due to noise, the equations can not be satisfied exactly). In this case, our goal is to minimize the sum of squares of residuals:
   $Sum_i((x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 - (t - t_i)^2) \to min.$
   You program should continuously read data from a virtual GPS receiver and print the position in each moment until the signal is lost. The number of satellites (and their order) can change. The initial position is unknown, but the position between successive readings does not change too much. The required accuracy is given by
   $Sum_i((x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 - (t - t_i)^2) < 10^{-6}$.
   It is guaranteed that such solution exists. The coordinates $x$; $y$; $z$ are in range [-10; 10], the time $t$ is in range [-1000; 1000].
   The number of readings is guaranteed not to exceed $10^5$.
2. You have to build a simulation software for a new chemical reactor. Your program is given a list of chemical reactions and initial concentrations of all components. You should output the concentrations after time $t$.
   In first-order reactions, only one molecule is necessary for the reaction, and the reaction rate is proportional to the concentration of this reagent:

$A \xrightarrow{k_1} n_1 B_1 + n_2 B_2 + ... + n_I B_I$.

For this reaction, we can write down the following system of ODEs:

$d[A]/dt = -k_1[A]$

$d[B_i]/dt = n_i k_1[A]$, $i = 1, ..., I$.

Here, $[A]$ is the concentration of molecule $A$, $[B_i]$ is the concentration of $B_i$ molecules, and $k_1$ is the reaction rate constant.

In second order reactions, two molecules are necessary for the reaction to proceed:

$A + C \xrightarrow{k_2} n_1 B_1 + n_2 B_2 + ... + n_I B_I$.

$d[A]/dt = -k_2[A][C]$

$d[B_i]/dt = n_i k_2[A][C]$, $i = 1, ..., I$.

3. The simplest example of oscillating chemical system is the Oregonator [http://www.scholarpedia.or
   which consists of the following reactions:

   $A + Y \xrightarrow{k_1} X + P$.

   $X + Y \xrightarrow{k_2} 2P$.

   $A + X \xrightarrow{k_3} 2X + 2Z$.

   $2X \xrightarrow{k_4} A + P$.

   $B + Z \xrightarrow{k_5} Y$.

   The reaction rates will always be within order of magnitude from their respective values in the example input file.

   Input

   The first line contains a single integer number $T = 1...1000$ – how long we will run our virtual reactor. The second line contains six floating-point values – the initial concentrations of $X$, $Y$, $Z$, $A$, $B$ and $P$. The third line contains five floating-point values – the reaction rate constants $k_1, ..., k_5$.

   Output

   The output should contains six floating-point values – the final concentrations of $X$, $Y$, $Z$, $A$, $B$ and $P$. Required precision is $10^-6$.

## K.49 Mathematical Modeling

- **Course name:** Mathematical Modeling
- **Course number:** XYZ
- **Knowledge area:** Mathematical Physics

### K.49.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 3rd year of BS
- **Semester of instruction:** 1st semester
- **No. of Credits:** 4 ECTS
- **Total workload on average:** 144 hours overall
- **Class lecture hours:** 2 per week
- **Class tutorial hours:** 2 per week
- **Lab hours:** 2 per week
- **Individual lab hours:** 0
- **Frequency:** weekly throughout the semester
- **Grading mode:** letters: A, B, C, D

### K.49.2 Prerequisites

- Mathematical Analysis I
- Mathematical Analysis II
- Analytic Geometry and Linear Algebra I
- Analytic Geometry and Linear Algebra II
- Discrete Math and Logic
- Differential Equations

### K.49.3 Course outline

The overall goal of this course is to enable students to build mathematical models of real-world systems, analyze them and make predictions about behavior of these systems. Variety of modeling techniques will be discussed with examples taken from physics, biology, chemistry, economics and other fields. The focus of the course will be on seeking the connections between mathematics and physical systems, studying and applying various modeling techniques to creating mathematical description of these systems, and using this analysis to make predictions about the system's behavior.

### K.49.4 Expected learning outcomes

After completing this course, a student will be able to:

- Assess and articulate what type of modeling techniques are appropriate for a given physical system.
- Construct a mathematical model of a given physical system and analyze it.
- Make predictions of the behavior of a given physical system based on the analysis of its mathematical model.

### K.49.5 Expected acquired core competences

- Concepts and practical use of mathematical modeling

### K.49.6 Textbook

-
-
-

### K.49.7 Reference material

NA

### K.49.8 Required computer resources

No special needs.

### K.49.9 Evaluation

- Weekly quizzes (20%)

- Home assignments (30%)
- Midterm Exam (20%)
- Final Exam (30%)

# K.50 Introduction to Computer Vision

- **Course name:** Computer Vision
- **Course number:** XXX

## K.50.1 Course Characteristics

### K.50.1.1 Key concepts of the class

- Computer vision techniques
- Classical and deep learning models

### K.50.1.2 What is the purpose of this course?

This course provides an intensive treatment of a cross-section of the key elements of computer vision, with an emphasis on implementing them in modern programming environments, and using them to solve real-world problems. The course will begin with the fundamentals of image processing and image filtering, but will quickly build to cover more advanced topics, including image segmentation, object detection and recognition, face detection, content-based image retrieval, artificial neural networks, convolutional neural networks, generative adversarial networks and much more. A key focus of the course is on providing students with not only theory but also hands-on practice of building their computer vision applications.

### K.50.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to process the video

- Significant exposure to real-world implementations
- To develop research interest in the theory and application of computer vision

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to choose the correct computer vision model.

- Suitability of different computer vision models in different scenarios
- Ability to choose the right model for the given task

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to deploy and developed models.

- Hands on experience to implement different models to know inside behavior
- Sufficient exposure to train and deploy model for the given task
- Fine tune the deployed model in the real-world settings

#### K.50.1.4 Course evaluation

**Table K.109:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 20 |
| Interim performance assessment | 30 | 50 |
| Exams | 50 | 30 |

#### K.50.1.5 Grades range

**Table K.110:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 90-100 |
| B. Good | 75-89 | 75-89 |
| C. Satisfactory | 60-74 | 60-74 |
| D. Poor | 0-59 | 0-59 |

#### K.50.1.6 Resources and reference material

- Handouts supplied by the instructor
- Materials from the interment and research papers shared by instructor
-
-

### K.50.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.111:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Image Acquisition and Basic Image Processing | 8 |
| 2 | Image Filtering and Binary Vision | 8 |
| 3 | Feature Extractors and Descriptors | 16 |
| 4 | Deep Learning models for computer vision | 16 |

#### K.50.2.1 Section 1

**Section title:** Image Acquisition and Basic Image Processing

**Topics covered in this section:**

- Computer vision in action
- The Human Vision System
- Optical Illusions
- Sampling and Quantization
- Image Representation
- Colour Spaces

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are the color spaces and where it's used?
2. What are the primary and secondary colors?
3. How image is formed into computers?
4. How you will convert the RGB to grayscale images

**Typical questions for seminar classes (labs) within this section**

1. Loading and plotting the images in python environment
2. Convertion of different color spaces
3. How you find the skin in the images based on the color space models
4. how to find red eye dot in face using color space models

**Test questions for final assessment in this section**

1. How you can distinguish different color spaces?
2. Explain and provide the reason for the blind spot creation in human eye.
3. In what scenarios computer vision is better than human vision?
4. Write down different robotic application areas where computer vision is applied successfully.

#### K.50.2.2 Section 2

**Section title:** Image Filtering and Binary Vision

**Topics covered in this section:**

- Image noise
- Convolutions and kernels
- Smoothing and blurring
- Thresholding and histograms
- Morphological operations
- Gradients and Edge detection

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are the challenges to perform histogram task?
2. Apply convolutional filter to calculate the response.
3. What kind of parameters are required to apply different image filters?
4. How you will compute the gradients of the image and its benefits?

**Typical questions for seminar classes (labs) within this section**

1. Implement Otsu Method
2. Implement Sobel, Preweitt filters
3. Implement Canny edge detector
4. Perform analysis over the different filtering on the given images

**Test questions for final assessment in this section**

1. Calculate the kernels for the given images
2. Explain the difference between different filters
3. What is image noise and how it contributes to make the computer vision task difficult?
4. Apply different combination of the filters to achieve the required output of the given image.

#### K.50.2.3 Section 3

**Section title:** Feature Extractors and Descriptors

**Topics covered in this section:**

- Histogram of Gradients (HoG)
- Scale-invariant feature transform (SIFT)
- Harris corner detector
- Template matching
- Bag of visual words
- Face Detection and Recognition (Viola Johns)

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. How feature extractor works over the given image?
2. What is the difference between the feature extraction and descriptors?
3. Explain the examples of descriptors and feature extractors.
4. Write down the pros and cons of SIFT, HOG and Harris.

**Typical questions for seminar classes (labs) within this section**

1. Implement template matching algorithm
2. Implement histogram of gradient using CV2 library
3. Implement of SIFT for the given task
4. Implement Harris corner detection
5. Analysis of different extractors for the given task

**Test questions for final assessment in this section**

1. How you distinguish different feature extractors and descriptors?
2. What are the possible methods to detect the corners?
3. How corners are useful to help the robotic vision task?
4. How you will patch the different images to construct the map of the location?

### K.50.2.4 Section 4

**Section title:** Deep learning models for computer vision

**Topics covered in this section:**

- You Only Look Once: Unified, Real-Time Object Detection (YOLO)
- Generative Adversarial Networks (GAN)
- Fully Convolutional Networks (FCN) for semantic segmentation

- Multi Domain Network (MDNet) for object tracking
- Generic Object Tracking Using Regression Networks (GOTURN) for object tracking

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How classification task is different from detection task?
2. Explain the transfer learning mechanism for object detection task.
3. How many types of model exist for object tracking in videos.
4. Write down the pros and cons of YOLO, FCN and MDNet.

**Typical questions for seminar classes (labs) within this section**

1. Implement YOLO using transfer learning mechanism
2. Implement GAN for MNIST dataset
3. Implement FCN and GOTURN
4. Analysis of different models for the given task

**Test questions for final assessment in this section**

1. What are the loss functions used in YOLO?
2. What are the learnable parameters of FCN for semantic segmentation?
3. How semantic segmentation is different from instance segmentation?
4. Write the application areas for object tracking in robotics.

# K.51 Microcontrollers and Embedded Hardware

- **Course name:** Microcontrollers and Embedded Hardware
- **Course number:**

## K.51.1 Course Characteristics

## K.51.2 What subject area does your course (discipline) belong to?

Logic circuits; Timing analysis; Robotic components; Real-time operating systems; Process management; Embedded software.

### K.51.2.1 Key concepts of the class

- Microcontroller and its peripherals function
- Program an embedded system
- Real-time applications

### K.51.2.2 What is the purpose of this course?

This subject focuses on the study of advanced microcontroller along with various applications using microcontrollers. It also briefs the students about interfacing of memory and various I/O devices like A to D converter, D to A converter LED, LCD to advanced microcontrollers. The students learn the Programming language (Embedded C) used for microcontrollers. They will be able to use the advanced fast microcontroller.

### K.51.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize

- The difference between Microprocessor and Microcontroller based system.
- Differences between: (i) Von-Neuman vs Harvard architecture (ii) RISC vs CISC architecture

- List of the special features of PIC18F452.
- The bit pattern of IPR1 & PIE1 registers of PIC18F452
- Arithmetic operation in PIC18F Family

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- How microcontroller and its peripherals function and interface to external peripherals
- The various challenges faced in embedded system design
- The Branching instructions of PIC18F microcontroller
- The Multiply and divide operations in PIC18F Family with example.
- The timers of PIC18F microcontroller and explain any one in brief.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Program an embedded system in assembly and C language
- Design, implement and test a single-processor embedded systems for real-time applications in engineering automation
- Optimize embedded software for speed and size for industrial applications

#### K.51.2.4 Course evaluation

**Table K.112:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/assignments | 25 | 45 |
| Interim project report | 25 | 15 |
| Final project report | 50 | 40 |

The course grades are given according to the following rules: Labs/assignments = 45 pts, Interim project report = 15 pts, Final project report = 40 pts.

**Table K.113:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 65-79 |
| C. Satisfactory | 60-74 | 40-64 |
| D. Poor | 0-59 | 0-39 |

#### K.51.2.5 Grades range

#### K.51.2.6 Resources and reference material

Main textbook:

- "Fundamentals of Microcontrollers and Applications in Embedded Systems" Ramesh Gaonkar.

Other reference material:

- “The 8051 Microcontroller and Embedded Systems Using Assembly and C,” Muhammad Ali Mazidi, Janice G Mazidi & Rolin McKinlay.
- “The 8051 Microcontrollers: Architecture, Programming and Applications” K Uma Rao,Andhe Pallavi.
- “Embedded systems architecture, programming and design, second edition” Raj Kamal.

### K.51.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.114:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Microcontroller and its peripherals function | 7 |
| 2 | Programming an embedded Systems | 8 |
| 3 | Real-time applications | 7 |

#### K.51.3.1 Section 1

**Section title:** Microcontroller and its peripherals function

**Topics covered in this section:**

- Basics of embedded System design
- Microcontroller based systems
- Historical perspective
- von Neumann versus Harvard Architecture
- CISC versus RISC Processors
- Interfacing real world devices with 8051 microcontroller

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Define Embedded systems. Give examples
2. Explain the logic operators in embedded C with examples.
3. Describe the hardware units of a generalized Embedded system.
4. Discuss the difference between Microprocessor and Microcontroller based system.

**Typical questions for seminar classes (labs) within this section**

1. Write a short note on Matrix Keyboard interfacing with 8051 microcontroller.
2. Explain serial data transmission and reception in 8051 microcontroller
3. Draw an interfacing diagram of DC motor with 8051 and explain in detail.
4. Define baud rate

**Test questions for final assessment in this section**

1. Differentiate between external hardware interrupts and serial communication interrupts
2. Explain the stack operations in 8051 microcontroller
3. Discuss De-bouncing problem of key and its solution

#### K.51.3.2 Section 2

**Section title:** Programming an embedded Systems

**Topics covered in this section:**

- C programming basics and time delay in 8051 C
- I/O programming in8051 C, Logic operations in 8051 C
- Accessing code ROM space in 8051 C
- Programming 8051 timers/Counter in C
- Basics of serial communication, 8051 connection to RS232
- 8051 interrupts programming in C
- DC motor interfacing and PWM using 8051.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. How can we interface 8051 with RS-232? Explain the importance of RI & TI flags.
2. Explain bit configuration of PIR1 of PIC18F452.
3. Discuss polling vs interrupts & enlist the various interrupts found in 8051.

**Typical questions for seminar classes (labs) within this section**

1. Draw & explain working of H-bridge circuit for DC motor interfacing with 8051.

2. WAP in C to toggle bit P2.7 every 20ms using Timer0.
3. Explain the stack operation in PIC18f with an example.

**Test questions for final assessment in this section**

1. Explain the interfacing of Relay with microcontroller.
2. Draw the internal structure of PORTA of 8051 microcontroller.
3. Write an Embedded C program for 8051 to find number of positive and negative data among ten byte of array. Send number of positive data to Port P3 and negative data to Port P2.

#### K.51.3.3 Section 3

**Section title:** Real-time applications

**Topics covered in this section:**

- Microcontroller Architecture – PIC18F Family
- PIC18F Family Programming model , Introduction to PIC18F Family instruction Set
- Data copy operations, Arithmetic operations, branching operations
- Stack and Subroutines and Illustrative Programs.
- Basics concepts of Input/output ports and Interfacing input/output Peripherals
- PIC18F Interrupts, PIC18F Timers, Illustration programs.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Describe how to handle an external interrupt give to a port pin in PIC18f series microcontroller. Discuss how you will write an Interrupt Service Routine in C language.

2. Explain the concept of current sinking in case of LED interfacing.
3. Explain the different branch instructions of PIC18F with examples

**Typical questions for seminar classes (labs) within this section**

1. Explain following instructions of 51 family microcontroller [1] MOV A,55h [2] MOV A,#55h [3] MOV @R0,#44h.
2. Write an ALP to clear 10 RAM locations starting at location 60H.
3. Draw & explain the memory organization in PiC18F4xx

**Test questions for final assessment in this section**

1. Write 8051 C programs to send temperature range of -4 to +4 to port P1.
2. List out the timers of PIC18F microcontroller and explain any one in brief
3. Write an 8051 C program to toggle all the bits of P0 and P2 continuously with a 200ms delay

## K.52 Distributed Ledgers Technologies

- **Course name:** Distributed Ledgers Technologies
- **Course number:** XYZ
- **Knowledge area:** Computer Science and Engineering

### K.52.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 4th year of BS
- **Semester of instruction:** 1st semester
- **No. of Credits:** 5 ECTS
- **Total workload on average:** 216 hours overall
- **Frontal lecture hours:** 2 per week.
- **Frontal tutorial hours:** 2 per week.
- **Lab hours:** 2 per week.
- **Individual lab hours:** 0.
- **Frequency:** weekly throughout the semester.
- **Grading mode:** letters: A, B, C, D.

### K.52.2 Course outline

Distributed Ledgers are the technologies on which cryptocurrencies, smart contracts, and all the blockchain-related technologies and applications are based on. Such technologies are an evolution of the ones at the base of the development of distributed systems and cryptographic technologies with a wide range of novel application domains. Moreover, the possibility of developing automated contracts (smart contracts) provides the tools to create distributed applications without any central control enabling a new development paradigm.

### K.52.3 Expected learning outcomes

- Understanding of the distributed consensus mechanisms
- Distributed computation and storage

- Ability to write smart contracts
- Understanding of the Bitcoin technology
- Understanding of the Ethereum technology

### K.52.4 Expected acquired core competences

- Smart contracts programming
- Development of a blockchain-based system
- Development of Dapps

### K.52.5 Prerequisites

- Principles of Computer Security
- Distributed Systems and Cloud Computing
- Network and Cybersecurity

### K.52.6 Detailed topics covered in the course

The course covers the following main topics:

- Fault-tolerance
- Consensus
- Quorum systems
- Blockchains architectures
- Distributed storage
- Cryptocurrences and Bitcoin
- Smart contracts and Ethereum

### K.52.7 Textbook

-
-

## K.52.8 Reference material

- Lecture and lab slides and material
- Several resources available online identified by the instructors during classes

## K.52.9 Required computer resources

Students should have laptops.

## K.52.10 Evaluation

- Mid-term Exam (30%)
- Final Exam (40%)
- Project (30%)

## K.53 Practical Machine Learning and Deep Learning

- **Course name:** Practical Machine Learning and Deep Learning
- **Course number:** PMLDL-04

### K.53.1 Course Characteristics

#### K.53.1.1 Key concepts of the class

- Practical aspects of deep learning (DL)
- Practical applications of DL in Natural Language Processing, Computer Vision and generation.

#### K.53.1.2 What is the purpose of this course?

The course is about the practical aspects of deep learning. In addition to frontal lectures, the flipped classes and student project presentations will be organized. During lab sessions the working language is Python. The primary framework for deep learning is PyTorch. Usage of TensorFlow and Keras is possible, usage of Docker is highly appreciated.

#### K.53.1.3 Course objectives based on Bloom's taxonomy

The course focuses on the following outcomes:

- Students should be able to apply deep learning methods to effectively solve practical (real-world) problems;
- Students should be able to work in data science team; to understand of principles and a lifecycle of data science projects.

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able

- to apply deep learning methods to effectively solve practical (real-world) problems;

- to work in data science team;
- to understand of principles and a lifecycle of data science projects.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able

- to understand modern deep NN architectures;
- to compare modern deep NN architectures;
- to create a prototype of a data-driven product.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able

- to apply techniques for efficient training of deep NNs;
- to apply methods for data science team organisation;
- to apply deep NNs in NLP and computer vision.

#### K.53.1.4 Course evaluation

**Table K.115:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | |
| Interim performance assessment | 30 | |
| Exams | 50 | |

If necessary, please indicate freely your course's features in terms of students' performance assessment.

#### K.53.1.5 Grades range

#### K.53.1.6 Resources and reference material

- Goodfellow et al. Deep Learning, MIT Press. 2017
- Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. 2017.

**Table K.116:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

- Osinga, Douwe. Deep Learning Cookbook: Practical Recipes to Get Started Quickly. O'Reilly Media, 2018.

## K.53.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.117:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Review. CNNs and RNNs | 12 hours of lectures, 12 hours of labs |
| 2 | Team Data Science Processes | 6 hours of lectures, 6 hours of labs |
| 3 | VAE, GANs | 8 hours of lectures, 8 hours of labs |

### K.53.2.1 Section 1

**Section title:** Review. CNNs and RNNs

**Topics covered in this section:**

- Image processing, FFNs, CNNs
- Training Deep NNs
- RNNs, LSTM, GRU, Embeddings
- Bidirectional RNNs
- Seq2seq
- Encoder-Decoder Networks
- Attention
- Memory Networks

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Suppose you use Batch Gradient Descent and you plot the validation error at every epoch. If you notice that the validation error consistently goes up, what is likely going on? How can you fix this?
2. Is it a good idea to stop Mini-batch Gradient Descent immediately when the validation error goes up?
3. List the optimizers that you know (except SGD) and explain one of them
4. Describe Xavier (or Glorot) initialization. Why do you need it?

Take the code from `https://github.com/lazyprogrammer/machine_learning_examples/blob/master/nlp_class3/attention.py`. Rewrite code in attention.py to:

1. Use GRU instead of LSTM;
2. Use pretrained Embedding weights in the Decoder;
3. Not use any attention at all.
4. Replace the softmax_over_time function with its Keras implementation.

**Typical questions for seminar classes (labs) within this section**

1. Name advantages of the ELU activation function over ReLU.
2. Can you name the main innovations in AlexNet, compared to LeNet-5? What about the main innovations in GoogLeNet and ResNet?
3. What is the difference between LSTM and GRU cells?

**Test questions for final assessment in this section**

1. Explain what the Teacher Forcing is.
2. Why do people use encoder–decoder RNNs rather than plain sequence-to-sequence RNNs for automatic translation?

3. How could you combine a convolutional neural network with an RNN to classify videos?

#### K.53.2.2 Section 2

**Section title:** Team Data Science Processes

**Topics covered in this section:**

- Team Data Science Processes
- Team Data Science Roles
- Team Data Science Tools (MLFlow, KubeFlow)
- CRISP-DM
- Productionizing ML systems

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is CRISP-DM?
2. What is TDSP?
3. How to use MLflow?
4. What is TensorBoard?
5. How to apply Kubeflow in practice?

**Typical questions for seminar classes (labs) within this section**

1. Explain issues in distributed learning of deep NNs.
2. How do you organize your data science project?
3. Recall a checklist for organization of a typical data science project.

**Test questions for final assessment in this section**

1. Can you explain what it means for a company to be ML-ready?
2. What a company can do to become ML-ready / Data driven?
3. Can you list approaches to structure DS-teams? Discuss their advantages and disadvantages.
4. Can you list and define typical roles in a DS team?
5. What do you think about practical aspects of processes and roles in Data Science projects/teams?

### K.53.2.3 Section 3

**Section title:** VAEs, GANs

**Topics covered in this section:**

- Autoencoders
- Variational Autoencoders
- GANs, DCGAN

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is an Autoencoder? Can you list the structure and types of Autoencoders?
2. Can you describe ways to train Stacked AEs?
3. What is Denoising AE? Can you describe what is sparsity loss and why it can be useful?
4. Can you make a distinction between AE and VAE?

**Typical questions for seminar classes (labs) within this section**

- If an autoencoder perfectly reconstructs the inputs, is it necessarily a good autoencoder? How can you evaluate the performance of an autoencoder?
- How do you tie weights in a stacked autoencoder? What is the point of doing so?
- What about the main risk of an overcomplete autoencoder?
- How the loss function for VAE is defined? What is ELBO?
- Can you list the structure and types of a GAN?
- How would you train a GAN?
- How would you estimate the quality of a GAN?
- Can you describe cost function of a Discriminator?

**Test questions for final assessment in this section**

- Can you make a distinction between Variational approximation of density and MCMC methods for density estimation?
- What is DCGAN? What is its purpose? What are main features of DCGAN?
- What is your opinion about Word Embeddings? What types do you know? Why are they useful?
- How would you classify different CNN architectures?
- How would you classify different RNN architectures?
- Explain attention mechanism. What is self-attention?
- Explain the Transformer architecture. What is BERT?

## K.54 Software Quality and Reliability

- **Course name:** Software Quality and Reliability
- **Course number:** XYZ

### K.54.1 Course characteristics

#### K.54.1.1 Key concepts of the class

- Quality Models and Metrics
- Verification Techniques and Testing
- Adequacy Criteria
- Reliability Engineering
- Cost of quality

#### K.54.1.2 What is the purpose of this course?

Building high-quality software is utmost important. However, it is easier said than done. The course is an overview of software quality and software reliability engineering methods. It includes introduction to software quality, overview of static analysis methods, testing techniques and reliability engineering. The students will put in practice the methods during laboratory classes and will dig down the topics in a small realistic project. The course balances traditional lectures, laboratory class and a small course project in which students apply the concepts and methods they learn to real artifacts. The course project consists in a quality analysis of an open source project of student's choice.

#### K.54.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students will remember:

- Several views on software quality.
- Trade-offs among quality attributes in quality models.
- Major differences verification techniques.

- Adequacy criteria for verification.
- Definition of reliability and the ways to calculate the necessary reliability.
- Cost of quality.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- Quality models usage.
- Cost of quality concept.
- Strengths and weaknesses of specific verification techniques.
- Reliability engineering.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Define a quality model of a software project in a given context.
- Select appropriate verification techniques and justify their adequacy.
- Define a necessary reliability for a software project in a given context.
- Justify quality related decisions to different stakeholders based on the cost of quality concepts.

#### K.54.1.4 Course evaluation

### K.54.2 Evaluation

**Table K.118:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 10 |
| Interim performance assessment | 30 | 50 |
| Exams | 50 | 40 |

The students performance will be evaluated as follows:

- Mid-term exam (20%)
- Final exam (20%)
- Reading Questions (10%)
- Project: mid-term presentation (20%)
- Project: final report (20%)
- Participation (10%)

#### K.54.2.1 Grades range

**Table K.119:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 65-79 |
| C. Satisfactory | 60-74 | 50-64 |
| D. Poor | 0-59 | 0-49 |

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table M.73, but it may change slightly (usually reduced) depending on how the semester progresses.

#### K.54.2.2 Resources and reference material

-
-
-
-
- Elena Dubrova, "Fault Tolerant Design"
- Laura L. Pullum, "Software Fault Tolerance Techniques and Implementation"
- Heiko Koziolek, “Operational Profiles for Software Reliability”
- G. D. Everett and R. McLeod, Jr, "Software Testing: Testing Across the Entire Software Development Life Cycle"
- Daniel Galin. "Costs of Software Quality"
- Stefan Wagner. "Software Quality Economics for Combining Defect-Detection Techniques"

## K.54.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.120:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Defining quality | 8 |
| 2 | Verification and Testing | 10 |
| 3 | Reliability | 6 |

### K.54.3.1 Section 1

**Section title:**

Defining quality

**Topics covered in this section:**

- Introduction, Views on Quality
- Quality Models
- Measurements & Quality Metrics
- Cost of quality

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the dominant quality view implicit in SCRUM and RUP?
2. Explain in your own words and in no more than three sentences the main contribution of one of the quality gurus like Ishikawa?
3. What is the difference between must have attributes and delighters in Kano's concept?

4. What is the main difference between a quality model like ISO 25010 and SAP Products Standard?
5. Describe in your own words and with regards to ISO 25010 the following quality attributes: Security, Reliability and Maintainability.
6. What is Kruchten's definition and taxonomy of Technical Debt?
7. According to Highsmith, what is relation of Technical Debt and Cost of Change?
8. In McConnell's taxonomy which type of Technical Debt can be positive?

**Typical questions for seminar classes (labs) within this section**

1. Define major quality focus by the customer in a given project.
2. Using SONAR evaluate maintainability of a given project.
3. Discuss you interpretation of the obtained quality level in a given project.
4. Describe how and what for quality models are useful? Provide an example from your studio project.
5. Map the requirement "the system shall be easy to maintain" to the ISO 25010 Quality model. Provide a definition to the metric level for at least two sub-characteristics for the requirement, and represent the mapping graphically.
6. Give an example of possible appraisal costs for a given project.
7. Present the quality model for the practicum project.

**Test questions for final assessment in this section**

1. Explain the difference between product quality, quality in use and process quality. Provide 2-3 quality attributes of each category briefly describing them.
2. What quality view best encompasses the phrase "Quality consists of the extent to which a specimen [a product-brand-model-seller combination] possesses the service characteristics you desire".
3. Explain the difference between accuracy and precision of measurement methods.
4. For each of the following quantities, indicate the scale (nominal, ordinal, interval, or ratio) of the data (just the scale, no justification required): a. Categories of defect types in a bug database. b. Branch coverage of a test suite. c. Severity of the defects in a bug database. d. Statement coverage of a test suite. e. Number

of features delivered on a milestone.

5. Explain the different types of technical debt that a project might incur.
6. Give a definition of constituent parts of the cost of quality.

#### K.54.3.2 Section 2

**Section title:**

Verification overview and Testing

**Topics covered in this section:**

- Verification Overview
- Measuring Tests Adequacy
- Input Domain Testing
- Random & Mutation Testing

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. In the context of mutation testing: a. What is an equivalent mutant? b. What is the meaning of the terms killed and dead on arrival? c. What is the difference between the two?
2. Develop BVA test cases for an application that implements the logic as defined in the exercise.
3. What is the relation between branch coverage and mutation testing?
4. What is an infeasible path?
5. What is fuzz testing? How it is different from random testing?
6. What is the oracle problem?

**Typical questions for seminar classes (labs) within this section**

1. Write a short code snippet that contains a possible null-pointer exception, and two different sets of test cases that achieve full branch coverage for the snippet. The first set of test cases should miss the defect; the second should trigger it.
2. Develop a classification tree covering all test relevant aspect for a Merge method. The method accepts two ordered integer vectors with a maximum of 128 elements each and returns a single ordered vector with no-duplicates formed from the elements of the input vectors.
3. Develop test cases for the logical function (A & B) | C -> D so that it achieves 100% MC/DC.
4. Develop test cases to achieve 100% basis path coverage utilizing McCabe method for the program below. Include: control flow graph, basis paths, test cases.

**Test questions for final assessment in this section**

1. Identify equivalence classes using Decision Table Method for a given problem.
2. Calculate number of test cases to achieve Basis Path coverage for a code sample.
3. Provide a test set that achieves full Basis Path coverage for a code sample.
4. Explain "dead on arrival" concept in the context of mutation testing.
5. Give examples of several types of usage for the fuzz testing.

#### K.54.3.3 Section 3

**Section title:**

Reliability

**Topics covered in this section:**

- Reliability Introduction and Necessary Reliability
- System Reliability and Reliability Strategies
- Operation Profile and Performance Testing

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Assume that a software system will experience 150 failures in infinite time. The system has now experienced 60 failures so far. The initial failure intensity at the

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

beginning of system testing was 15 failures per CPU hour. What is the current failure intensity?

2. Explain in your words the check-point recovery mechanism. Assuming that several communicating processes run in parallel, what will be a limitation of the checkpoint recovery mechanism?
3. In your words, define the Operational Profile.
4. Explain impact on Utilization on Response Time.
5. Explain Amdahl's law relation to performance improvements during development process.

**Typical questions for seminar classes (labs) within this section**

1. For the course project define the necessary reliability.
2. Explain in your words the difference between N-version programming and N-version self-checking programming.
3. Explain how the operational profile concept can be applied to regression and load tests.
4. From a pure performance point of view is it better, the same or worst to have a single server or two with half the speed?
5. You execute a benchmark test twice and find that the performance of the system was 30 transactions/hour the first time and 20 transactions/hour the second time. What is the average throughput?
6. You execute a load test for one hour, first during peak hour and again off-peak. During peak hour the system process 20 transactions/hour. Off-peak it processes 30 transactions/hour. What is the average throughput?

**Test questions for final assessment in this section**

1. Assume that a software system is undergoing system-level tests and the initial failure intensity is 15 failures per CPU hour. The failure intensity decay parameter has been found to be 0.025 per failure. So far the test engineers have observed 60 failures. What is the current failure intensity?
2. Explain in your own words the role of the voter in N-version methods.
3. Describe in your words limitations of Operational Profile.
4. Give an example illustrating general relationships between response time, throughput, and resource utilization.

## K.55 Mobile Application Development

- **Course name:** Mobile Application Development
- **Course number:** XYZ
- **Knowledge area:** Computer Science and Engineering

### K.55.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 4th year of BS
- **Semester of instruction:** 1st semester
- **No. of Credits:** 5 ECTS
- **Total workload on average:** 216 hours overall
- **Frontal lecture hours:** 2 per week.
- **Frontal tutorial hours:** 2 per week.
- **Lab hours:** 2 per week.
- **Individual lab hours:** 0.
- **Frequency:** weekly throughout the semester.
- **Grading mode:** letters: A, B, C, D.

### K.55.2 Course outline

Mobile technologies are becoming ubiquitous and used for diverse purpose such as business, industry, education, entertainment and medical. In this course, students will practice with popular mobile application development environments and methods to create mobile applications. The course is designed for undergraduate students of the Software Engineering track to cover aspects that are peculiar to the development of applications on mobile computing platforms and that have not been covered by other courses. Android is the reference for teaching programming techniques and design patterns related to the development of standalone applications and mobile interfaces to enterprise and cloud systems. The course will cover processes, tools and frameworks that are required to develop applications for mobile computing devices.

### K.55.3 Expected learning outcomes

- Ability to apply general programming knowledge in the field of developing mobile applications
- Describing and comparing different mobile application models, architectures and patterns
- Applying mobile application models, architectures and patterns to the development of a mobile software application
- Describing the components and structure of a mobile development framework (Google's Android Studio)

### K.55.4 Expected acquired core competences

- Mobile App design
- Construct software for a mobile application
- Prototyping
- Graphic design for mobile devices
- Android

### K.55.5 Prerequisites

- Introduction to Programming I
- Introduction to Programming II
- Data Modelling and Databases I
- Data Modelling and Databases II
- SE Project
- Software Architecture

### K.55.6 Detailed topics covered in the course

The course covers the following main topics:

- Android
- Graphic design for mobile devices

### K.55.7 Textbook

- 
- 

### K.55.8 Reference material

- Lecturing and lab slides and material will be provided
- Several resources are available online and will be pointed during the course

### K.55.9 Required computer resources

Students should have laptops and possibly mobile devices

### K.55.10 Evaluation

- Mid-term Exam (30%)
- Final Exam (40%)
- Project (30%)

## K.56 Mechatronics

- **Course name:** Mechatronics
- **Course number:**

### K.56.1 Course characteristics

### K.56.2 What subject area does your course (discipline) belong to?

Sensors and actuators; Robotic control; Electromechanical systems.

### K.56.3 Key concepts of the class

- System design
- Feedback control
- Electric motor selection and control

### K.56.4 What is the purpose of this course?

Industrial requirements for modern mechatronics and robotics engineers are such that, given an automation task, they must be able to identify and model the targeted physical process or system, select appropriate sensors, actuators and transmission mechanisms to control it, and design and implement control algorithm to ensure the system achieves desired performance. Therefore, the purpose of this class is to familiarize the students with the most fundamental aspects of all the key areas described above while focusing on typical real-world exercises and examples.

### K.56.5 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize

- Various types of sensors, their pros and cons,
- Working principles of electric actuators (DC motors),
- Operation principles of motion transmission mechanisms,

- Fundamentals of linear feedback control systems, and
- Principles of controller design for mechatronic systems.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- How to select sensors for a given application,
- How to choose appropriate transmission mechanisms and account for their efficiency,
- How to integrate all selected parts to create a mechatronic system,
- Typical nonlinearities that originate from electronic and mechanical sources and their effects on system performance, and
- How to tune control system for selected motor and desired performance specifications.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Drive differential equations of motion describing behavior of physical systems with several degrees of freedom,
- Calculate motor and sensor requirements for a given physical system, control task or application,
- Select appropriate motor that provides enough power while avoiding overheating,
- Tune control system for selected motor, transmission mechanism and sensor to achieve desired response and stability.

### K.56.6 Course evaluation

If necessary, please indicate freely your course's features in terms of students' performance assessment:

The course grades are given according to the following rules: Homework assignments (4) = 20 pts, Quizzes (4) = 40 pts, Term project = 40 pts.

**Table K.121:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 0 |
| Interim performance assessment | 30 | 60 |
| Exams | 50 | 40 |

### K.56.7 Grades range

**Table K.122:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

If necessary, please indicate freely your course's grading features.

### K.56.8 Resources and reference material

Main textbook:

- "Mechatronics," Sabri Cetinkunt. John Wiley & Sons., 2007.

Other reference material:

- "System dynamics: modeling, simulation, and control of mechatronic systems," Dean C. Karnopp, Donald L. Margolis, and Ronald C. Rosenberg. John Wiley & Sons, 2012.
- "Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering" (6th Edition), William Bolton. Pearson Education, 2003.

### K.56.9 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Dynamics and electrodynamics | 6 |
| 2 | Electric motors | 6 |
| 3 | Transmission mechanisms and sensors | 4 |
| 4 | Control systems | 6 |

## K.56.10 Section 1

### Section title:

Dynamics and electrodynamics

### Topics covered in this section:

- Free body motion
- Kinetic and potential energy
- Differential equations of motion
- Basics of linear electric circuits
- Impedance

### What forms of evaluation were used to test students' performance in this section?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

### Typical questions for ongoing performance evaluation within this section

1. Find kinetic and potential energy of a given physical system.
2. Write differential equations of motion of a mechanical system.
3. Find system transient response when external force is applied.
4. Solve for voltages and currents in a given electric circuit.
5. Find electric power produced by individual circuits' components.

### Typical questions for seminar classes (labs) within this section

1. In MATLAB or other software, do the following for a second-order system:
    - Write a program to simulate its dynamics;

- Plot step response and find its parameters;
- Calculate and plot system energy with time.

2. Find and analyze frequency response (Bode plot) of a dynamic system.
3. Find impedance of a given linear electric circuit.
4. Write differential equations describing voltages and currents of dynamic circuit.
5. How does electric energy exchange in an RLC circuit?

**Test questions for final assessment in this section**

1. Use Kirchoff's voltage and current laws to drive differential equations of a given electric circuit.
2. What are the analogies between three main mechanical (mass, damper, spring) and electrical (resistance, capacitance, inductance) components?
3. Find impedance of electrical circuit and draw a corresponding mechanical system schematically.

## K.56.11 Section 2

**Section title:**

Electric motors

**Topics covered in this section:**

- Electric and magnetic fields
- Operating principles of DC motors
- Electromechanical dynamic model of DC motors
- Steady-state torque-speed characteristics, power
- AC motors
- Linear motors
- Energy losses in electric motors

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. What devices around us are based on principles of electromagnetism?
2. Describe what happens when an conductive wire is placed in magnetic field.

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

3. How to create a rotating magnetic field, and what happens to magnetic objects placed inside of it?
4. How to find electrical and mechanical power of electric motor?
5. What happens when a motor is running in the generator mode?

**Typical questions for seminar classes (labs) within this section**

1. Drive differential equations governing the motion of a DC motor.
2. Draw a block diagram of a DC motor based on differential equations and convert them into state-space form.
3. Calculate maximum DC motor speed in no-load and loaded conditions.
4. Assuming a DC motor, calculate stall torque and maximum torque for given speed.

**Test questions for final assessment in this section**

1. Draw a schematic diagram of electrical and mechanical parts of DC motor and explain their interplay.
2. What is the physical meaning of mechanical time constant of a DC motor? What about electrical time constant? How to calculate them?
3. Explain how to select a DC motor based on known force-velocity profile of the application.

## K.56.12 Section 3

**Section title:**

Transmission mechanisms and sensors

**Topics covered in this section:**

- Rotary-to-rotary transmission
- Rotary-to-translational motion transmission mechanisms
- Shaft misalignments and flexible couplings
- Position sensors
- Velocity and acceleration sensors
- Force and torque sensors

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What applications of gears do you know? What about belts and pulleys?
2. What are the possible sources and effects of shaft misalignments on electric motors and gears?
3. Give an example of sensors used in conventional home appliances.
4. Typical applications where we need velocity and acceleration measurements.
5. Given a particular real-world device (system), which transmission mechanisms and sensors does it use in your opinion?

**Typical questions for seminar classes (labs) within this section**

1. What are potential drawbacks and nonlinearities of conventional motion transmission mechanisms?
2. What effect does a transmission mechanism have on required motor speed and torque?
3. What are the pros and cons of each conventional position sensor type?
4. Name applications where position measurement is not feasible.
5. What are the main issues of conventional force-torque sensors?

**Test questions for final assessment in this section**

1. Derive differential equations of motion when a load is connected to DC motor shaft via transmission and analyze its contribution to overall system dynamics.
2. For an application of motion control with desired accuracy and given a DC motor, select appropriate transmission mechanism and sensor resolution to satisfy performance specifications.
3. List pros and cons of conventional position sensor types and briefly describe their preferable application areas.

### K.56.13 Section 4

**Section title:**

Control systems

**Topics covered in this section:**

- Feedback control systems
- Stabilization and trajectory tracking
- Linear regulators (P, PD, PID)
- Digital control (sampling, quantization)
- DC motor controller tuning
- Stability of dynamic systems

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Give examples of real-world biological and physical closed-loop (feedback) systems
2. Drive error dynamics equations for a given feedback control law

3. What are the physical analogies of each term in PID-controller?
4. How to implement PD regulator in MATLAB software?

**Typical questions for seminar classes (labs) within this section**

1. In MATLAB, simulate behavior of a linear second-order ODE for the following controller types: P, PD, PID.
2. Analyze stability of a given feedback control system.
3. How does sampling affect system stability?
4. Tune controller for a given motor-transmission-sensor combination.

**Test questions for final assessment in this section**

1. What is the physical analog of PD-regulator in application to control over second-oder mechanical systems?
2. What are the effects of time delays, quantization, and sampling rates on stability of digital control systems?
3. For a given application, select DC motor and tune its control system.

## K.57 Business Analytics

- **Course name:** Business Analytics
- **Course number:** XYZ
- **Knowledge area:** xxx

### K.57.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 4th year of BS
- **Semester of instruction:** 1st semester
- **No. of Credits:** 4 ECTS
- **Total workload on average:** 144 hours overall
- **Class lecture hours:** 2 per week
- **Class tutorial hours:** 2 per week
- **Lab hours:** 2 per week
- **Individual lab hours:** 0
- **Frequency:** weekly throughout the semester
- **Grading mode:** letters: A, B, C, D

### K.57.2 Prerequisites

- Introduction to Programming I
- Introduction to Programming II
- Probability and Statistics
- Data Mining

### K.57.3 Course outline

In this course, students will learn about using data about customers and markets in business decision-making. They will learn to identify, evaluate, and capture business analytic opportunities that create value. Toward this end, students will learn how to gather, analyze, and interpret data about markets and customers. They will also explore the challenges that can arise in implementing analytical approaches within an organization.

### K.57.4 Expected learning outcomes

- Understanding of how decision-makers use business analytics to formulate and solve business problems
- Understanding and applying the appropriate tools for the the analysis of quantitative and qualitative data
- Fluency with the use of software packages for data analysis
- Understanding data gathering
- Analyzing and interpreting outputs (graphs, tables, mathematical models, etc.)
- Understanding how to collect data and report results objectively

### K.57.5 Expected acquired core competences

- Data Mining and data management techniques
- Data Visualization
- Descriptive Analytics
- Predictive Analytics
- Prescriptive Analytics
- Decision Analytics

### K.57.6 Textbook

-

### K.57.7 Reference material

- Lecturing and lab slides and material will be provided
- Several resources are available online and will be pointed during the course

### K.57.8 Required computer resources

Students should have laptops.

### K.57.9 Evaluation

- Assignments and project (30%)
- Mid-term Exam (30 %)
- Written Final (40%)

## K.58 English for Academic Purposes I

- **Course name:** English for Academic Purposes-1
- **Course number:** N/A
- **Subject area:** English for Academic Purposes (EAP)

### K.58.1 Course Characteristics

### K.58.2 What subject area does your course (discipline) belong to?

English for Specific Purposes, English for Academic Purposes, Academic Writing

#### K.58.2.1 Key concepts of the class

- Academic discourse
- Argumentation in higher education

#### K.58.2.2 What is the purpose of this course?

The course is designed to equip the first-year undergraduate students of the Department of Computer Science (CS) with the necessary English language knowledge and skills for their effective performance in CS subjects, by means of better comprehension of academic texts, and by expressing and supporting their viewpoints orally and in writing, in the manner appropriate in the academic environment.

#### K.58.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize

- Academic vocabulary
- Cohesive devices

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- Writing Process
- Structure and purpose of descriptive, compare-contrast, cause-effect, persuasive paragraphs
- The concepts of coherence and cohesion
- What plagiarism is and strategies to avoid it
- IMRaD structure and purpose
- Structure and purpose of a 5 paragraph reading response essay and its parts

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Read, comprehend and annotate a popular scientific article, identifying main ideas and supporting details
- Identify and articulate personal responses on what has been read
- Select and utilize appropriate paragraph structure and wording for a particular purpose
- Utilize tools to avoid plagiarism – paraphrasing, quoting, and citing
- Utilize in-text citation and organize a reference list according to a specified referencing style
- Self-edit and peer review of the written production and effectively manage the writing process
- Outline and write a coherent and concise 5 paragraph reading response essay
- Express and support their points of view in a paired and individual academic presentation
- Express and support their points of view in an academic discussion
- Give constructive feedback and suggestions on peer verbal and written performance

#### K.58.2.4 Course evaluation

**Late Submission Policy**

This policy will be strictly applied in this course. If a personal emergency should arise that affects your ability to turn in an assignment in a timely fashion, you must

**Table K.123:** Course grade breakdown

| Task | Weight | Task Details |
|---|---|---|
| Assignment (1) | 5 | Sentence and Paragraph Structure Quiz |
| Assignment (2) | 20 | Paragraph Writing (in-class) |
| Assignment (3) | 5 | Academic Vocabulary Quiz |
| Assignment (4) | 30 | Reading Response Essay Writing (in-class) |
| Assignment (5) | 30 | Oral Presentation |
| Timely submission | 10 | Timely Submission of Assignments throughout the Course |
| Participation | 10 | Physical Attendance and Active Participation in the Course |

contact the course instructor BEFORE the deadline to get a "Special Late Submission Approval" from the course instructor. Documents that prove the urgency of your situation must be submitted to your instructor, e.g., health reports. Without the "Special Late Submission Approval", submissions will be still accepted up to 48 hours late, but with a 4 25submissions should be submitted via LMS.

**Cooperation Policy and Quotations**

We encourage vigorous discussion and cooperation in this class. You should feel free to discuss any aspects of the class with any classmates. However, we insist that any written material that is not specifically designated as a Team Deliverable be done by you alone. This includes answers to reading questions, individual reports associated with assignments, and labs. We also insist that if you include verbatim text from any source, you clearly indicate it using standard conventions of quotation or indentation and a note to indicate the source.

#### K.58.2.5 Grades range

**Table K.124:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 86,25-100 |
| B. Good | 75-89 | 61,25-86,24 |
| C. Satisfactory | 60-74 | 36,25-61,24 |
| D. Poor | 0-59 | 0-36,24 |

#### K.58.2.6 Resources and reference material

Main textbook:

- The materials are designed from a variety of textbooks and authentic sources to meet the students' own needs and interests.

Other reference material:

- Cambridge Academic English Intermediate, Craig Thaine et al., CUP, 2012
- Cambridge Academic English Upper Intermediate, Martin Hewings et al., CUP, 2014.
- Cambridge Academic English Advanced, Martin Hewings et al., CUP, 2012
- Academic Writing Skills 1, Peter Chin et al., CUP, 2014
- cademic Writing Skills 2, Peter Chin et al., CUP, 2012.
- Skills for Effective Writing Level 4, CUP, 2013.
- Skills for Effective Writing Level 3, CUP, 2013.
- Final Draft Level 3, Andrew Aquino-Cutcher et al., CUP, 2016.
- Final Draft Level 4, Wendy Asplin et al., CUP, 2016.
- Writing Around the World: A Guide to Writing Across Cultures, Matthew McCool, 2009. https://apastyle.apa.org/

### K.58.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.125:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | From sentence to paragraph | 16 |
| 2 | Reading and annotating academic texts | 12 |
| 3 | Reading-response essay | 20 |
| 4 | Academic public talk | 12 |

#### K.58.3.1 Section 1

**Section title:** From sentence to paragraph

**Topics covered in this section:**

- Sentence structure. Simple, complex and compound sentences
- Coherence and Cohesion. Cohesive devises within sentences
- Paragraph structure
- Writing Process
- Paragraph types - descriptive, compare-contrast, cause-effect, persuasive paragraphs

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | No |
| Homework and group projects | Yes |
| Midterm evaluation | Yes |
| Testing (written or computer based) | Yes |
| Reports | No |
| Essays | No |
| Oral polls | No |
| Discussions | No |

**Typical questions for ongoing performance evaluation within this section**

1. Write sentences of a particular structure and type
2. Identify and name cohesive devises
3. Outline paragraph. Provide relevant details to support your claim
4. Write paragraphs of particular structure and type, with relevant supporting details
5. Write a piece of reflective writing about your learning experience

**Typical questions for seminar classes (labs) within this section**

1. On a quiz, demonstrate your knowledge of paragraph structure and types
2. Write impromptu a well-structured paragraph of a particular type within limited time, using appropriate cohesive devices and providing relevant supporting details.
3. Perform self-editing
4. Perform peer review and provide peer feedback for a piece of writing
5. Elaborate on your personal strategy for improving your writing
6. Write a second draft of your text based on peer and mentor's feedback

**Test questions for final assessment in this section**

1. On a closed-books test, write impromptu a well-structured and concise paragraph of a particular type using appropriate cohesive devices, and providing relevant supporting details

#### K.58.3.2 Section 2

**Section title:** Reading and annotating academic texts

**Topics covered in this section:**

- Reading purposes and strategies
- Academic vocabulary
- IMRaD structure
- Identifying main ideas and reading for details
- Annotating texts
- Expressing your opining on what was read, supporting your opinion

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Read and understand an article
2. Annotate an article
3. Identify academic vocabulary

**Typical questions for seminar classes (labs) within this section**

1. In a group discussion, express your opinion on what has been read, and provide relevant support
2. Identify and utilize academic vocabulary in speech

**Test questions for final assessment in this section**

1. On a quiz, demonstrate your skill of understanding and utilization of academic vocabulary

#### K.58.3.3 Section 3

**Section title:** Reading-response essay

**Topics covered in this section:**

- Structure of a basic 5 paragraph essay. Coherence and cohesion between paragraphs
- Plagiarism. Strategies to avoid plagiarism
- APA referencing style

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Outline an essay. Articulate clear thesis statement with two claims, and provide relevant support for those claims
2. Write introductory, body and concluding paragraphs of the essay.

3. Use paraphrasing, summarising and citations as strategies to avoid plagiarism
4. Use APA referencing style

**Typical questions for seminar classes (labs) within this section**

1. Write introductory, body and concluding paragraphs of the essay
2. Perform peer review of a piece pf writing and provide feedback
3. Perform self-editing of a piece of writing
4. Elaborate on your personal strategies to improve you writing, base on self- peer- and mentor feedback

**Test questions for final assessment in this section**

1. Write a well-structured and concise 5 paragraphs essay with a reference list, providing relevant supporting details and avoiding plagiarism. Use APA referencing style.

### K.58.3.4 Section 4

**Section title:** Academic public talk

**Topics covered in this section:**

- Academic public talk structure
- Public speech vocabulary
- Dealing with QA

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Public talk | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Outline your public talk
2. Anticipate questions for QA session
3. Practice your public talk and use appropriate transitional expressions and vocabulary Avoid plagiarism

**Typical questions for seminar classes (labs) within this section**

1. Practice your public talk and QA session.
2. Perform peer review of a public talk and provide constructive feedback
3. Perform self-evaluation of your video-recorded public talk
4. Elaborate on your personal strategies for improving your public speaking skills

**Test questions for final assessment in this section**

1. Perform a well-structured public talk, articulating and supporting your point of view with relevant details. Avoid plagiarism. Answer your audience questions.

**Assessment Task (2): Paragraph Writing**

**Type:** Individual, in-class, and paper/computer-based assignment

**Length:** 2 paragraphs, each paragraph different type, each paragraph 70-130 words

**Rationale:** This task prepares you to demonstrate your understanding of the structure of an effective paragraph; the topic sentence, supporting ideas, and concluding sentence. In addition, you have to demonstrate your ability to develop your paragraph with sufficient explanations, details and examples. Furthermore, this task will help you self-edit and revise your own work, and provide peer-feedback on your colleagues' work.

**Instructions:** Provide two types of paragraphs (descriptive and comparative) on the same topic. Familiarize yourself with the assignment grading scheme. Refer to the Moodle materials that are relevant to the paragraph module. Consider time for revising and editing your work.

**Duration:** 90 minutes

**Weight:** 20 %

| **Criterion** | **A** | **B** | **C** | **D** |
|---|---|---|---|---|
| Task Response (25%) | Fully responds to all parts of the task, and develops relevant to the purpose, concise, clear and well supported paragraphs. | Addresses all parts of the task although some parts may be more fully covered than others; presents relevant main ideas but some may be unclear and /or repetitive. | Addresses the task only partly; although presents some main ideas but these are limited and not sufficiently developed; there may be irrelevant detail and redundant words. | Does not adequately address any part of the task; irrelevant to the purpose; presents few ideas which are largely undeveloped or irrelevant |
| Topic and Concluding Sentences Quality (15%) | In both paragraphs: original topic sentence, reflecting thought and insight; focused on one main idea; appropriate and summative concluding sentence that elaborates the controlling idea of the topic and effectively indicates the closure of the paragraph. | In both paragraphs: clearly stated topic sentence presents one main idea; a concluding sentence that refers to the controlling idea of the topic and/or summaries most ideas, a good indication of the closure of the paragraph | In both paragraphs: acceptable topic sentence presents one idea; acceptable concluding that attempts to refer to the controlling idea of the topic and/or summaries some ideas, an attempt to indicate the closure of the paragraph. | In both paragraphs: poorly attempts to state the topic sentence to present one idea poorly attempts to state the concluding sentence with no clear reference to the controlling idea of the topic, with no indication to the main ideas, poorly attempts to indicate the closure of the paragraph. |
| Sentences (15%) | In both paragraphs: concrete and descriptive examples and persuasive and original supporting details with thorough, relevant and qualitative explanations. | In both paragraphs: examples and details relate to the topic and some explanation is included. Good quality, but not sufficient support. | In both paragraphs: examples and details poorly support the topic. The quality of examples and explanations is superficial. | In both paragraphs: lack of/irrelevant examples and details, no attempts made to provide any support. |
| Coherence and Cohesion (15%) | In both paragraphs: sequences information and ideas logically; there is a clear progression throughout; uses a range of cohesive devices appropriately. | In both paragraphs: arranges information and ideas coherently; uses cohesive devices effectively but cohesion within and/or between sentences may be faulty or mechanical; may not always use referencing clearly or appropriately. | In both paragraphs: presents information with some organization but there may be a lack of referencing and substitution. | In both paragraphs: does not organize ideas logically; poor usage of cohesive devices and those used may not indicate a logical relationship between ideas. |

| | | | | |
|---|---|---|---|---|
| Lexical Resource (15%) | In both paragraphs: uses a wide range of vocabulary fluently and efficiently to convey precise meanings; produces rare errors in spelling and/or word formation. | In both paragraphs: uses only a limited range of structures; attempts complex sentences but these tend to be less accurate than simple sentences; may make frequent and grammatical errors and punctuation may be faulty; errors may cause some difficulty for the reader. | In both paragraphs: uses a limited range of vocabulary, but this is minimally adequate for the task; may make noticeable errors in spelling and/or word formation that can cause some difficulty for the reader. | In both paragraphs: uses only basic vocabulary and a limited range of words and expressions with very limited control of word formation and/or spelling; errors cause strain for the reader. |
| Grammatical Range and Accuracy (15%) | In both paragraphs: uses a variety of complex structures; the majority of sentences are error free; has good control of grammar and punctuation but may make a few errors. | In both paragraphs: uses a mix of simple and complex sentence forms; makes some errors in grammar and punctuation but they rarely reduce communication. | In both paragraphs: uses only a limited range of structures; attempts complex sentences but these tend to be less accurate than simple sentences; may make frequent and grammatical errors and punctuation may be faulty; errors can cause some difficulty for the reader. | In both paragraphs: uses only a very limited range of structures with only rare use of subordinate clauses; some structures are accurate, but errors predominate and distort the meaning; punctuation is mostly faulty. |

Adapted from: `https://takeielts.britishcouncil.org/sites/default/files/2018-01/IELTS_task_2_Writing_band_descriptors.pdf`

**Assessment Task (4): Reading Response Essay (RRE)**

**Type:** Individual & in-class and paper/computer-based assignment

**Length:** 5/6-paragraph essay - 450 words minimum

**Rationale:** This task prepares you to demonstrate your clear understanding of the RRE structure in which you respond to articles you have read. RRE should cover the following:

- Introductory paragraph with data from the original source (the title of the article and author), background information on the topic and your opinion (Thesis statement) on the author's ideas;

- Summary paragraph, containing the main ideas of the original article, examples and supporting information, in the order they appear in the source, without any personal comments;
- Body/Response paragraphs, 2-3 paragraphs, explaining your opinion/reaction/feelings to the ideas of the original article. Provide evidence upon defending your opinion and reference the sources you refer to by using the agreed upon (APA/IEEE) citation style; Support your ideas with sufficient explanation, details and examples.
- Concluding sentence, where you give credit to the author of the article, restate your thesis statement, and write a final thought sentence. This task will help you self-edit and revise your own work, and provide peer-feedback on your colleagues' work.

**Instructions:** Familiarize yourself with the assignment grading scheme. Refer to the Moodle materials that are relevant to the RRE module. Consider time for revising and editing your work.

**Duration:** 2 hours

**Weight:** 30 %

| Criterion | A | B | C | D |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Task Response 35% | The essay has all the five paragraphs in accordance with the RRE structure, with clear Introductory and Concluding paragraphs. The RRE summary clearly identifies the author's thesis and paraphrases the major points in the author's argument very well and shows a sophisticated analysis of the author's ideas. The response offers clear and concise reasons why the writer may agree or disagree with the author or how this article has increased the writer's understanding of the topic. The essay contains a Works Cited list, which includes all the sources mentioned in the essay. | The essay may not strictly follow the five-paragraph structure of a RRE essay but the writing may have some flows and not have very clear logic at times. Minor elements of the Introductory and Concluding paragraphs are missing. The RRE summary identifies the author's thesis and outlines most of the article(s) but may be missing a portion of the argument; analysis of the ideas is not very sophisticated and includes some of the primary evidence that the author has utilized to prove the thesis but may leave out other examples for exploring the thesis. The response develops one or two points very well. The works cited list has minor mistakes. | The essay follows the five-paragraph structure of a RRE essay but there is an imbalance between the paragraphs. In addition, the writing has flows of organization and does not have very clear logic at times. Elements of the Introductory and Concluding paragraphs are missing. The RRE summary identifies quite well the author's thesis and outlines most of the article(s) but may be missing some of the argument; analysis of the ideas is somehow superficial and includes some of the primary evidence that the author has utilized to prove the thesis but leaves out other examples for exploring the thesis. The response develops one or two points well. The works cited list has a few mistakes. | The essay does not follow the five- paragraph structure of a RRE essay, but even if it does, there is a great imbalance between the paragraphs. The writing has big flows of organization and its logic is difficult to follow. Introductory and Concluding paragraphs are poorly developed with a lot of essential information missing. The RRE summary does not identify the author's thesis and does not outline most of the article(s) lack of clarity of arguments and very simple language; analysis of the ideas is very superficial and includes poor evidence of what the author has utilized to prove the thesis. Despite the presence of a Works Cited List, the essay lacks essential elements and cannot be classified as a RRE. |
| Coherence 15% | Logically organizes information and ideas; there is clear progression throughout where ideas are concise and to the point. There are clearly defined 5 paragraphs. | Arranges information and ideas coherently and there is a clear overall progression, but ideas are not always presented in a concise manner. | Presents information with some organization but there may be a lack of overall progression and in terms of conciseness, the essay is much more descriptive than concise. | Presents information and ideas but these are not arranged coherently and there is no clear progression in the response. Logic of the essay is very superficial. |
| Cohesion 15% | Uses a range of cohesive devices appropriately. Presents a clear central topic within each paragraph. | Uses cohesive devices effectively, but cohesion within and/or between sentences may be faulty or mechanical | Makes inadequate, inaccurate or over- use of cohesive devices. May be repetitive because of the lack of referencing and substitution. | Uses some basic cohesive devices but these may be inaccurate or repetitive. May not write in paragraphs or their use may be confusing. |

| | | | | |
|---|---|---|---|---|
| Use of language 15% | Uses a sufficient range of vocabulary and less common, but academic lexical items with some awareness of flexibility in word formation and use of collocation.<br>Uses a variety of complex sentence structures, produces frequent error-free sentences. Has good control of grammar and punctuation. | Uses an adequate range of vocabulary for the task. Attempts to use less common vocabulary but with some inaccuracy. Makes some errors in spelling and/or word formation, but they do not impede communication. Uses a mix of simple and complex sentence forms. Makes some errors in grammar and punctuation but they rarely reduce overall communication | Uses a limited range of vocabulary, and it is sometimes irrelevant to the task. May make noticeable errors in spelling and/or word formation that may cause difficulty for the reader in comprehending the text. Uses a limited range of sentence structures, attempts to write complex sentences but these tend to be less accurate than simple sentences. May make frequent grammatical errors and punctuation may be faulty; errors can cause some difficulty for the reader. | Uses only basic/limited vocabulary which may be used repetitively, or which may be inappropriate for the task. Has limited control of word formation and/or spelling and errors may cause strain for the reader. Uses a very limited range of sentence structures with only rare use of subordinate clauses. Some structures are accurate, grammatical errors are predominant, and punctuation is often faulty. |
| Avoiding Plagiarism Strategies 10% | Student has excellent paraphrasing of the information from the article analyzed. Any information coming directly from the source is clearly quoted, there are in-text referencing and author tagging. | Student has done very good to paraphrasing of the information coming from the original text. In case of using original text information, it is properly quoted in most cases, there are good in- text referencing and author tags. | Student has tried to paraphrase the information necessary to sustain his/her ideas but is not always successful. In-text referencing is not very well done, quotation do not include the full quote and author tags and mainly missing. | Student has plagiarized a lot, poor efforts to paraphrase the information, poor in- text citations, few or no author tags. |
| Referencing 10% | Student shows a consistent and appropriate referencing style throughout the essay and in the reference list. | Student shows a consistent and appropriate referencing style in most of the essay and the reference list. | Student is quite consistent in using the appropriate referencing style in his/her the essay, the reference list might be missing. | Student is unclear on the proper referencing style to be applied in his/her the essay, the reference list is missing. |

**Assessment Task (5): : Oral Presentation – Expressing your opinion on what has been read**

**Type:** Paired public talk

**Rationale:** This task prepares you to express and support your opinion on what you have read, work collaboratively, manage your time effectively, and respond to what you have read critically by providing evidence.

**Instructions:** Select one of the given topics, read critically, annotate and summarise, respond to what you have read, and structure your presentation by expressing your opinion and supporting your ideas with evidence. Dealing with the questions is a focal part of your presentation. Familiarize yourself with the assignment grading scheme. Refer to the Moodle materials that are relevant to the OP module.

**Duration:** 15 min including QA

**Weight:** 20 %

| Criterion | A | B | C | D |
|---|---|---|---|---|
| Task Response-Effort - complexity of thought; in-depth preparation evidence 30% | The team has fully covered all the requirements of the task by: giving an effective public speech through which they demonstrate full comprehension of the already read article/s by showing clear signs of thorough understanding of the reference text/s. | The team has covered most of the requirements of the task by: giving a good public speech through which they demonstrate good comprehension of the already read article/s, showing signs of general understanding of the reference text/s. responding to ideas in the reading/s by expressing personal opinions and providing evidence that has supported most opinions. Evidence has been not reasonable enough to convince the listener. | The team has covered some of the requirements of the task by: giving a public speech through which they demonstrate fair comprehension of the already read article/s by showing no fully clear signs of understanding the reference text/s. partially responding to ideas in the reading/s by expressing personal opinions. the lack of evidence in most parts of the support and/or providing inappropriate evidence that does not contribute to the function of supporting opinions. | The student has not covered most of the requirements of the task, which are either missing or faulty. The students gives a public speech through which they demonstrate limited evidence of their comprehension of the already read article/s, the student poorly attempts to express personal opinion/s or reaction to an idea/s in the reading/s. the lack of evidence throughout the whole speech. |
| Referencing 15% | The sources are clearly and correctly referred to in both the verbal speech and the visuals (slides and/or handouts). | The sources are referred to and missed in either the verbal speech or the visuals (slides and/or handouts). | The sources are mentioned with faulty details in one/both of the verbal speech and the visuals (slides and/or handouts). | The lack of references in both the verbal speech and the visuals (slides and/or handouts). |

| | | | | |
|---|---|---|---|---|
| Presentation -overall impression - structure, pace, voice, interaction with the audience, body language, visuals, time management 25% | The content is organized logically with smooth transitions throughout the presentation. The opinion on the article read is clear, logically analysed and properly supported with personal opinions /views on the subject. Moves through their speech at an appropriate pace. Makes pauses at the end of sentences or at significant moments. Speaks clearly and forcefully throughout their entire speech. Consistently maintains eye contact with the audience throughout their speech and with everybody presented in the room. Body language, gestures, and facial expressions add greatly to the message. Visual aids well-chosen and presented in support of the student's personal reaction to a read text. Speech delivered within allotted time. Very good time management of the presentation and question time. | The organization of the content is congruent; transitions are smooth. Opinion on the text read is well expressed but supporting ideas are a bit broad. Usually moves through their speech at an appropriate pace. Makes most pauses at the end of sentences or at significant moments. Speaks clearly and forcefully throughout most of speech. Mostly maintains eye contact with the audience throughout their speech, but sometimes with only a part of the room. Body language, gestures, and facial expressions compliment message. Visual aids well-chosen and presented. Speech delivered within allotted time. Good time management of the presentation and question time. | The organization of the content shows some planning of personal opinion and support of it; but transitions are not smooth. Maintains appropriate pace only in some parts of their speech. Pauses at the end of sentences are much more for word search then for thought completion. Speaks relatively clearly and forcefully throughout some parts of their speech. Mostly maintains eye contact with the audience throughout their speech, but with limited sections of the room. Body language, gestures, and facial expressions compliment message. Minor problems with visual aids. Speech delivered within 30% of allotted time. Fair time management of the presentation and question time. | Shows little planning. Introduction, body, conclusion poorly organized. Moves through their speech too quickly or too slowly. Speech is either very soft or too loud for most of the presentation. As a result, lack of attention of the audience. Little or no eye contact. Body language, facial expressions and gestures lack variety and spontaneity. Significant problems with visual aids. Speech delivered within 20% of allotted time. Poor time management of the presentation and question time. |
| Dealing with questions (answering questions) 15% | Can respond effectively and spontaneously at normal speed. Can present a convincing argument and sustain his/her personal opinion or view.Responds accurately to all questions. | Can present a convincing argument and sustain an opinion or view. Responds to most questions but sometimes hesitates in his/her answers. | Can maintain a question-answer session with reasonable accuracy, even when not always finding the appropriate words. | Responds with short choppy answers which do not always correspond to question. |

| | | | | |
|---|---|---|---|---|
| Use of language (vocabulary and grammar range and accuracy, pronunciation, fluency) 15% | IELTS band 6 and above: Is prepared and willing to speak at length, though may lose coherence at times due to occasional repetition, self- correction or hesitation. Uses a range of connectives and discourse markers but not always appropriately. Has a wide enough vocabulary to explain and support their opinion at length and make meaning clear despite some inappropriate descriptions of ideas or personal opinion. Generally paraphrases successfully. Uses a mix of simple and complex structures, but with limited flexibility. May make frequent mistakes with complex structures, though these rarely cause comprehension problems. Uses a range of pronunciation features with mixed control. Can generally be understood throughout, though mispronunciation of individual words or sounds reduces clarity at times. | IELTS band 5 and 5,5: Usually maintains flow of speech but uses repetition, self-correction and/or slow speech to keep going. May overuse certain connectives and discourse markers. Produces simple speech fluently, but more complex communication causes fluency problems. Manages to present his/her opinion on the article read but uses vocabulary with limited flexibility. Attempts to use paraphrase but with mixed success. Produces basic sentence forms with reasonable accuracy. Uses a limited range of more complex structures, but these usually contain errors and may cause some comprehension problems. | IELTS band 4 and 4.5 : Cannot present his/her opinion without noticeable pauses and may speak slowly, with frequent repetition and self-correction. Links basic sentences but with repetitious use of simple connectives and some breakdowns in coherence. Is able to talk about the ideas in the article read but can only convey their basic meaning without clearly formulating an opinion on them. Makes frequent errors in word choice. Rarely attempts paraphrase. Produces basic sentence forms and some correct simple sentences but subordinate structures are rare. Errors are frequent and may lead to misunderstanding. Uses a limited range of pronunciation features. Attempts to control features but lapses are frequent. Mispronunciations are frequent and cause some difficulty for the listener. | IELTS band 3 and 3.5: Speaks with long pauses. Has limited ability to link simple sentences. Gives only simply formulated personal opinion and is frequently unable to provide adequate support of it. Uses simple vocabulary to convey the meaning of the read information. Attempts basic sentence forms but with limited success or relies on apparently memorised utterances. Makes numerous errors except in memorised expressions. Presentation is difficult to follow. |

| **Criterion** | **A** | **B** | **C** | **D** |
|---|---|---|---|---|
| Participation | Contributes often and voluntarily. Works well with others. Insightful comments. | Contributes readily, works well with others, makes useful comments. | Contributes occasionally voluntarily, works well with others, makes sensible comments. | Contributes reluctantly, does not always work well with others, comments sometimes irrelevant. |
| Attendance | 80% + | 70%+ | 50%+ | 50%- |

| Quizes | A | B | C | D |
|---|---|---|---|---|
| Success rate | 90% + | 75%+ | 60%+ | 59%- |

| Criterion | A | B | C | D |
|---|---|---|---|---|
| Timely submission | 90% + | 70%+ | 50%+ | 30%- |

## K.59 English for Academic Purposes II

- **Course name:** English for Academic Purposes-2
- **Course number:** N/A
- **Subject area:** English for Academic Purposes (EAP)

### K.59.1 Course Characteristics

### K.59.2 What subject area does your course (discipline) belong to?

English for Specific Purposes, English for Academic Purposes, Academic Writing

#### K.59.2.1 Key concepts of the class

- Academic discourse
- Argumentation in higher education

#### K.59.2.2 What is the purpose of this course?

The course is designed to equip the first-year undergraduate students of the Department of Computer Science (CS) with the necessary English Language knowledge and skills for their effective performance in CS subjects, by means of better comprehension of academic texts, and by expressing and supporting their viewpoints orally and in writing, in the manner appropriate in the academic environment.

### K.59.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize

- Academic vocabulary for academic style, discussions and debates
- Scientific papers search databases
- Rhetorical devices

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- The process of academic information seeking
- The notions of credible and non-credible sources
- The purpose of documenting the process of academic information search
- The purpose and structure of a reading log and annotated bibliography
- The concepts related to academic writing style: normalization, hedging, wordiness and redundancy, over-generalization
- The general rules of applying IEEE referencing style for in-text citations and reference list
- Structure and purpose of a 5 paragraph research essay and its parts
- General principles of rhetoric and constructing arguments

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Perform targeted academic literature search using appropriate peer-reviewed databases
- Identify credible sources
- Manage and document the reading process with a search log and annotated bibliography
- Demonstrate the ability to read critically - to be able to evaluate the strengths and limitations of ideas and approaches

- Comprehend academic articles and judge their relevance to your areas of interest
- Work independently and in groups to manage the ongoing and recursive processes of drafting, editing and revising, writing, and incorporating the instructor's feedback and peer review comments
- Give impromptu and prepared written feedback to peers, receive impromptu and prepared written feedback from your instructor and peers, and incorporate this new understanding into your revised work
- Use IEEE in-text citations and referencing
- Produce a well-structured coherent, cohesive and concise 5-paragraph essay, using appropriate advanced academic language and style, supporting your relevant ideas with research-based evidence
- Develop and support your points of view in academic discussions, interviews and debates with strong and valid evidence
- Perform academic discussions in different roles and consequently develop a deeper knowledge of the target research question and make group decisions
- Give constructive impromptu and prepared oral feedback to peers. Receive impromptu and prepared oral feedback from your instructor and peers, reflect on both types of feedback and incorporate this new understanding into your final work
- Develop strong argumentation and use specialized debate vocabulary and rhetorical devices to present your resolution on a specified topic, to defend it, and to convince the audience

#### K.59.2.4 Course evaluation

**Late Submission Policy**

This policy will be strictly applied in this course. If a personal emergency should arise that affects your ability to turn in an assignment in a timely fashion, you must contact the course instructor BEFORE the deadline to get a "Special Late Submission Approval" from the course instructor. Documents that prove the urgency of your situation must be submitted to your instructor, e.g., health reports. Without the "Special Late Submission Approval", submissions will be still accepted up to 48 hours

**Table K.129:** Course grade breakdown

| Task | Weight | Task Details |
|---|---|---|
| Assignment (1) | 10% | Reading Log (individual submission) |
| Assignment (2) | 10% | Annotated Bibliography (group submission) |
| Assignment (3) | 10% | Interview (Group work, in-class) |
| Assignment (4) | 20% | Research Essay (individual submission) |
| Assignment (5) | 10% | Improving your Writing Skills Quiz (in-class) |
| Assignment (6) | 20% | Debate (group in-class) |
| Timely submission | 10% | Timely Submission of Assignments throughout the Course |
| Participation | 10% | Physical Attendance and Active Participation in the Course |

late, but with a 25% penalty. No "Special Late Submission Approval" will be granted after the deadline. All late submissions should be submitted via LMS.

#### Cooperation Policy and Quotations

We encourage vigorous discussion and cooperation in this class. You should feel free to discuss any aspects of the class with any classmates. However, we insist that any written material that is not specifically designated as a team deliverable needs to be done by you alone. This includes answers to reading questions, individual reports associated with assignments, and labs. We also insist that if you include verbatim text from any source, you clearly indicate it using standard conventions of quotation or indentation and a note to indicate the source.

#### K.59.2.5 Grades range

**Table K.130:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 86,25-100 |
| B. Good | 75-89 | 61,25-86,24 |
| C. Satisfactory | 60-74 | 36,25-61,24 |
| D. Poor | 0-59 | 0-36,24 |

#### K.59.2.6 Resources and reference material

Main textbook:

- The materials are designed from a variety of textbooks and authentic sources to meet the students' own needs and interests.

Other reference material:

- Cambridge Academic English Intermediate, Craig Thaine et al., CUP, 2012
- Cambridge Academic English Upper Intermediate, Martin Hewings et al., CUP, 2014.
- Cambridge Academic English Advanced, Martin Hewings et al., CUP, 2012
- Academic Writing Skills 1, Peter Chin et al., CUP, 2014
- cademic Writing Skills 2, Peter Chin et al., CUP, 2012.
- Skills for Effective Writing Level 4, CUP, 2013.
- Skills for Effective Writing Level 3, CUP, 2013.
- Final Draft Level 3, Andrew Aquino-Cutcher et al., CUP, 2016.
- Final Draft Level 4, Wendy Asplin et al., CUP, 2016.
- Writing Around the World: A Guide to Writing Across Cultures, Matthew McCool, 2009.
- `http://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE-Reference-Guide.pdf`

### K.59.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table K.131:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Academic information seeking and documenting | 22 |
| 2 | Research Essay | 22 |
| 3 | Debates | 16 |

#### K.59.3.1 Section 1

**Section title:** Academic information seeking and documenting

**Topics covered in this section:**

- Credible and non-credible sources
- Documenting targeted academic information seeking with a reading log and an annotated bibliography
- Academic reading - understanding graphs and tables
- IEEE for citations in annotated bibliographies

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Explain the characteristics of a credible source.
2. List and explain the types of academic information search.
3. Find a research-based article which is relevant to your research question and which is a credible source. Document it in your reading log.
4. Reflect on your search and group work process and elaborate on those processes improvement.
5. As a group, write a draft of an Annotated bibliography (AB), using IEEE.

**Typical questions for seminar classes (labs) within this section**

1. During a group discussion, brief your peers on the article of your choice, prove that it is a credible source and explain how it is relevant to you research question. As a group, decide if the article is appropriate for a 3-item AB.
2. Explain the meaning of a graph or a table.
3. Perform a round table discussion to account for your sources choices for AB.
4. Perform peer review of AB and provide peer feedback for a piece of writing.
5. Elaborate on your personal strategy for improving your AB.

**Test questions for final assessment in this section**

1. Individually submit your reading log that should contain at least 3 entries and should critically evaluate them in terms of relevance and credibility.
2. As a group, submit your annotated bibliography that contains at least 3 entries describing sources that are credible and relevant to your research question. Use IEEE for citations.
3. As a group, perform an interview and demonstrate your collaborative deep analysis of the researched area.

### K.59.3.2 Section 2

**Section title:** Research Essay

**Topics covered in this section:**

- Purpose and structure of a 5 paragraph research essay
- Academic writing style - nationalisation, hedging, wordiness and redundancy, over-generalization.
- IEEE for in-text citations. IEEE reference list.

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Search for research-based evidence to support the arguments for your essay
2. Outline your essay.
3. Draft introductory, body, concluding paragraphs of your essay.
4. Self-edit your essay text according to the structure checklist, academic style requirements checklist, and IEEE referencing check-list.

**Typical questions for seminar classes (labs) within this section**

1. Design a list of arguments in favour or against a point of view.
2. Demonstrate the research-based evidence to support those arguments.
3. Identify the cases of nominalization, charged language, wordiness and redundancy and over-generalisation and improve the texts.
4. Based on the mentor's review of your text, elaborate on your further steps to improve your writing.

**Test questions for final assessment in this section**

1. On a quiz, demonstrate your skill of understanding and utilization of academic style and dealing with nominalization, charged language, wordiness and redundancy and over-generalisation
2. Individually, submit your 5 paragraph research essay, referenced according to IEEE.

### K.59.3.3 Section 3

**Section title:** Debates

**Topics covered in this section:**

- Debates purpose and structure
- Debates vocabulary
- Rhetorical devices. Principles of rhetoric

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions (debates) | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Take a quiz on debates vocabulary
2. Take a quiz on rhetoric devices
3. Perform research for your arguments
4. Based on the mentor 's and peer feedback, elaborate on your personal strategies to improve your debate skills

**Typical questions for seminar classes (labs) within this section**

1. Construct and present your argumentation
2. Perform a mock debate
3. Perform a peer review of your group-mates debate

**Test questions for final assessment in this section**

1. Perform a debate group providing strong support for your arguments in order to convince your audience, and using appropriate vocabulary and rhetoric devices.

**Assessment Task (1): Information search documentation**

**Type:** Individual, submission

**Rationale:** This task prepares you to write your annotated bibliography assignment by keeping a record of what you have read during your academic search and critical reading of the academic sources you have selected to respond to your research question.

**Instructions:** You should document three sources at least of what you read during your academic search. These sources have to be relevant to your research question that has been selected by your search team. The following elements should be covered in your documentation/RL:

- Research Question (the same for the three sources/entries);
- Date of access (For electronic sources);

- Databases (Example: Google Scholar, ResearchGate, ScienceDirect, ACM Digital Library, IEEE, etc.)
- Date of publication;
- Source link;
- Author(s)/institution(s) 's (full name);
- Name of the source;
- Key terms/key words;
- Notes: What is the text about? What are the main concepts covered?;
- Methodology & Results used;
- Relevance of the source and the significance of certain information in the source to your research question.

You can add any extra elements to the above list if you think this element can be useful for your work.

**Submission deadline:** This task is submitted as a home assignment (not in-class) and it can take any form: table, flash cards, power point, etc. You should read the source, document it and have it ready for discussion on the due days referred to in the above calendar: RL1- lesson 4, RL2 Lesson 6, RL3- Lesson 7. The three submitted RLs/entries can be edited to produce a clear and comprehensible input and for writing the AB assignment. RL document due - Lesson 10

**Weight:** 10 %

| Criterion | A | B | C | D |
|---|---|---|---|---|
| Task response 50% | At least three entries of the documentation clearly cover the elements listed in the task description, concisely brief the main concepts of the sources, and deeply assess and evaluate the significance of the source to the research question. | Two of the three required entries of the documentation clearly cover the elements listed in the task description; concisely brief the main concepts of the sources, and deeply assess and evaluate the significance of the source to the research question. | One of the three required entries of the documentation clearly covers the elements listed in the task description; concisely briefs the main concepts of the sources, and deeply assesses and evaluates the significance of the source to the research question. | None of the entries of the documentation clearly covers the elements listed in the task description; concisely briefs the main concepts of the sources, and deeply assesses and evaluates the significance of the source to the research question. |
| Source relevance 20% | At least three cited sources indicate the relevance to the research question | Two of the cited sources indicate the relevance to the research question | One of the cited sources indicates the relevance to the research question | None of the cited sources indicates the relevance to the research question |
| Timely Submission 30% | At least three sources of the reading log are submitted on time (prior to the source discussion session) | Two sources of the reading log are submitted on time (prior to the source discussion session) | One source of the reading log is submitted on time (prior to the source discussion session) | None of the three sources is submitted on time (prior to the source discussion session) |

#### K.59.3.4 Assessment Task (2): Annotated Bibliography (AB)

**Type:** Written group work (three members)

**Length:** 600 words minimum

**Rationale:** This task prepares you to analyze academic sources, for performing your own research activity during your Computer Science studies at the university and further, in Research and Development (RnD) field.

**Instructions:** Provide an annotated bibliography (AB) containing 3 relevant and credible (peer reviewed, research-based) sources. Please refer to the AB template provided during the module to structure your work. The content of AB should cover the following items:

- Your research question
- Three cited sources according to IEEE reference style
- A summary and rationale of each source

You should cite your sources according to the Institute of Electrical and Electronics Engineers (IEEE) style: `http://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE-Reference-Guide.pdf` Your points should be substantive and clear and should use theoretical terminology from the articles.

**Weight:** 10 %

| Criterion | A | B | C | D |
|---|---|---|---|---|
| Task fulfillment and quality of annotations 40% | Task fulfillment is nearly perfect, with almost no errors of substance, detail or development. Annotations sufficiently and succinctly summarize, evaluate, and indicate the significance of the source to the research question. | Task fulfillment includes a few errors with a slight impact on substance, detail and development. Annotations sufficiently summarize, evaluate, indicate the significance of the source to the research question. | Assignment includes many errors which diminish or confuse substance, detail and development. Annotations adequately summarize, evaluate, indicate the significance of the source to the research question. | Assignment shows only a little or no task fulfillment. Some annotations provide an inadequate summary, evaluation, and/or indicate the significance of the source to the research question. |
| Language 20% :grammatical accuracy, clarity and formality | Language is very clear, effective, carefully chosen and precise, with a high degree of accuracy in grammar, vocabulary and sentence construction; register and style are effective and appropriate. | Language is clear and carefully chosen, with a good degree of accuracy in grammar, vocabulary and sentence construction; register and style are consistently appropriate. | Language is clear and carefully chosen, with an adequate degree of accuracy in grammar, vocabulary and sentence construction despite some lapses; register and style are mostly appropriate. | Language is sometimes clear and carefully chosen; grammar, vocabulary and sentence construction are fairly accurate, although errors and inconsistencies are apparent; the register and style are to some extent appropriate. |
| Credibility of sources 20% | The three sources cited can be considered reliable and trustworthy. | Two of the sources cited can be considered reliable and trustworthy. | One of the sources cited can be considered reliable and trustworthy. | No sources cited can be considered reliable. |
| Referencing 20% | Citations are formatted correctly in the document according to IEEE reference style. | There are a few formatting errors. Some citation information may be missing according to IEEE reference style. | There are some formatting errors or missing information according to IEEE reference style. | There are many and/or frequent formatting or information errors according to IEEE reference style. |

#### K.59.3.5 Assessment Task (3): Interview

**Type:** Group work (three members)

**Duration:** 15 minutes (with equal involvement of each team member)

**Rationale:** This task aims at highlighting your critical thinking skills after collaboratively analyzing the source articles you have searched for your written AB. Furthermore, you will be able to express your vision on what you read, in your Computer Science study.

**Instructions:** During this research discussion, you should verbally demonstrate your full understanding of the area of interest you have researched during your annotated bibliography module. Furthermore, your group interview should represent your deep analysis of the research area, your knowledge and reflective thoughts as a team on the questions attached to the task:

1. Is your article a credible source, what can determine its credibility?
2. What does your article revolve around?
3. How is your article relevant to your research question? Can you give examples?
4. How has the group collaboration been during the AB process?

This task can also be given as a presentation provided that each of the team members can respond to the aforementioned task questions and/or any relevant questions.

**Weight:** 10 %

| **Criterion** | **A** | **B** | **C** | **D** |
|---|---|---|---|---|
| Task response and quality of analysis and responding to questions 40% | Shows excellent knowledge and understanding of the source articles and their relevance to the research question.Able to explain any concepts clearly and answer the questions aligned to the task or any questions convincingly. | Shows good knowledge and understanding of the source articles and their relevance to the research question. Can explain most concepts and answer most questions aligned to the task or any questions convincingly. | Some gaps in knowledge and in understanding of the source articles and their relevance to the research question.Some difficulties in explaining concepts and answering the questions aligned to the task or any questions convincingly. | Poor understanding of the source articles and their relevance to the research question. Unable to explain concepts or answer most questions aligned to the task and any questions convincingly. |
| Language 25% : grammatical accuracy, clarity and formality | Language is very clear, effective, carefully chosen and precise, with a high degree of accuracy in grammar, vocabulary and sentence construction; register and style are effective and appropriate. | Language is clear & carefully chosen, with a good degree of accuracy in grammar, vocabulary & sentence construction; register and style are mostly appropriate. | Language is clear and carefully chosen, with an adequate degree of accuracy in grammar, vocabulary and sentence construction despite some lapses; the register and style are to some extent appropriate. | Language is sometimes clear and carefully chosen; grammar, vocabulary and sentence construction are fairly accurate, although errors and inconsistencies are apparent; the register and style are inappropriate. |
| Delivery and communication 15% : voice, gestures and tone, clarity | Controlled and effective use of voice, face, gesture, and tone in all cases. Audibility Excellent.Extremely fluent. The listener has no difficulty understanding the speaker's ideas at any time. | Controlled and effective use of voice, face, gesture, and tone in most cases, but with a few problems. Clearly audible. No strain to the listener. Communicates ideas with a good degree of fluency. | Controlled and effective use of voice, face, gesture, and tone in some cases with many problems. Audibility poor.Struggles to communicate meanings successfully.A certain amount of strain to the listener. | Deficient or absent delivery that shows only a little or no task fulfilment. Frequently inaudible. Considerable strain to the listener to understand the meaning. |
| Time Management 5% | As a group: Accurate time management: time is divided evenly among the team members. | As a group: Minor time management issues: A clear attempt to divide time evenly among the team members. | As a group: Poor time management: One or two of the team members either might either monopolize the discussion or not be mostly engaged throughout. | As a group: Lack of time management: all team members might either monopolize the discussion or not be engaged throughout. |
| Group collaboration 15% | Each member of the team shows excellent knowledge & understanding of their role in fulfilling the assignment and can explain any teamwork process clearly and answer any questions convincingly and evidently. | Each member of the team shows good knowledge and understanding of their role in fulfilling the assignment and can explain any teamwork process and answer most questions convincingly and evidently. | Each member of the team shows fair knowledge and understanding of their role in fulfilling the assignment and can explain most stages of the teamwork process and answer some questions convincingly and evidently. | Some gaps in the knowledge & understanding of the team member's role in fulfilling the assignment. Some difficulties in explaining the teamwork process and convincingly answering the questions and providing evidence. |

**Assessment Task (4): Research Essay**

**Type:** Individual work

**Rationale:** This task teaches you to express a position on a topic and use peer-reviewed researched-based sources to support this position

**Instructions:** Write a 5-paragraph essay on a specific topic describing your position on this topic and providing 3 fact-based arguments to support it. The choice of the topic to write about is yours. It can be a topic in any area where research content is used. Use 5-6 academic sources in order to provide arguments which are based on facts and details gathered from research. Include a reference list.

**Weight:** 20 %

| Criterion | A | B | C | D |
|---|---|---|---|---|
| Content 30% : information, analysis, task fulfillment | An exceptional depth, detail and development of ideas within and across paragraphs with almost no errors. | A good depth, detail and development of ideas within and across paragraphs, with only a few errors. | A satisfactory depth,detail and development of ideas within and across paragraphs, with several problems. | The assignment shows only a little or no task fulfillment. |
| Structure 25% : within paragraphs, across paragraphs,layout | The student has followed the suggested 5-paragraph structure. Information and ideas are logically organized throughout. The student has used appropriate structuring devices | The student has followed the suggested 5-paragraph structure, but there are some flaws. In most cases, information and ideas are logically organized throughout, structuring devices are used effectively. | The student has followed the suggested 5-paragraph structure, but there are serious flaws. There are many problems with the organization and progression of information and ideas, and with the use of structuring devices. | The student has not followed the suggested 5paragraph structure. Most structural devices are deficient or absent. Information and ideas are not arranged coherently. The student may not have written in paragraphs. |
| Language 20%:grammar, vocabulary | A variety of complex grammar structures and advanced level vocabulary are used throughout the essay. Sentences are error free. Punctuation and spelling are nearly perfect. | Most sentences are clear and error free, with a mix of simple and complex sentence forms. Some advanced level words are used. | Many sentences contain errors. A limited range of structures and advanced level vocabulary are used, with many errors which cause comprehension problems for the reader. | A limited range of complex sentence structures and vocabulary are used. Most structures are inaccurate. Comprehension is difficult. |
| Academic Writing Style 15% | The student has followed Academic Writing Style rules. The student has opted for nominalization where necessary and avoided overgeneralization, redundancy, charged language and wordiness. | The student has followed most rules of the Academic Writing Style, though some mistakes have been made. The student has not always opted for nominalization where necessary or avoided overgeneralization, redundancy, charged language and wordiness. | The student has violated many rules of the Academic Writing Style and failed to use nominalization where necessary or to avoid overgeneralization, redundancy, charged language and wordiness. | The student is not aware of the Academic Writing Style |
| Citing and referencing 10% | In-text citations and the References List are correctly formatted according to the IEEE style. All the necessary citation information is provided. | There are a few in-text citation and References List errors in the essay. Some citation information may be missing. | There are many in-text citation and References List errors in the essay. A lot of citation information is missing. | The student is not aware of the IEEE referencing style. |

## Assessment Task (5): Quiz

**Weight:** 10 %

| Quiz | A | B | C | D |
|---|---|---|---|---|
| Success rate | 90% + | 62%+ | 37%+ | 36%- |

**Timely submission**

**Weight:** 10 %

| Criterion | A | B | C | D |
|---|---|---|---|---|
| Timely submission | Timely submits 90%+ of the assignments | Timely submits 65%+ of the assignments | Timely submits 40%+ of the assignments | Timely submits 40%- of the assignments |

**Participation and attendance**

**Weight:** 10 %

| Criterion | A | B | C | D |
|---|---|---|---|---|
| Participation | Contributes often and voluntarily. Works well with others. Insightful comments. | Contributes readily, works well with others, makes useful comments. | Contributes occasionally voluntarily, works well with others, makes sensible comments. | Contributes reluctantly, does not always work well with others, comments sometimes irrelevant. |
| Attendance rate | 90% + | 65%+ | 40%+ | 40%- |

**Assessment Task (6): Debate**

**Type:** Group work (three members per team)

**Duration:** 15 minutes

**Rationale:** This task will prepare you to develop and support your points of view in formal debates with strong and valid research-based evidence. What is more, you will get training in using debate vocabulary and rhetorical devices to present your resolution on a specified topic, to defend it, and to convince the audience.

**Instructions:** You will choose a debate topic and form two teams – an affirmative (proposition) team and a negative (opposition) team. Each team will consist of 3 students. Each student will have to prepare a 1-minute speech presenting a well-grounded pro or con argument. Each team will then have 3 minutes to rebut all the arguments presented by the other team and to defend your team's resolution. During the rebuttal, each team member has a 1-minute speaking limit. If a speaker exceeds their 1-minute speaking limit, they will remain silent during the rest of the rebuttal. When speaking, you will have to use debate phrases and rhetoric devices as per the vocabulary list provided. During the debate, the audience will be present. The audience will serve as a jury deciding a debate winner. If two thirds of the audience vote for a team, such team wins, and the team gets extra 0.4 added to the assignment 6 grade (those 0.4, weighted accordingly, in some cases will be added to the final course grade score).

**Weight:** 20 %

| Criterion | A | B | C | D |
|---|---|---|---|---|
| Statement 20% | Controlled and effective reasoning and evidence in all cases. All arguments are strong, persuasive, clearly tied to an idea and organized in a tight, logical fashion | Controlled and effective reasoning and evidence in most cases, but with a few problems. Most arguments are rather strong, quite persuasive, clearly tied to an idea and organized in a tight, logical fashion | Controlled and effective reasoning and evidence in some cases with many problems. Some of the arguments are irrelevant | Faulty reasoning and evidence show a little or no task fulfillment. Most of the arguments are either irrelevant and illogical, or altogether absent. |
| Delivery 29% | Controlled and effective use of voice, face, gesture, body language and tone in all cases. | Controlled and effective use of voice, face, gesture, body language and tone in most cases, but with a few problems. | Controlled and effective use of voice, face, gesture, body language and tone in some cases with many problems. | Deficient or absent delivery that shows only a little or no task fulfillment. |
| Language 20% | Sentences are almost error free. An effective range of vocabulary, and a variety of structures and terms are used. | Most sentences are error free. An effective range of vocabulary, and a variety of structures and terms are used. | Many sentences contain errors. A limited range of vocabulary and structures are used, with many errors which cause listening comprehension problems. | A limited range of structures and complex sentence forms are used. Most structures are inaccurate, and errors predominate. Comprehension is difficult. |
| Rhetoric 20% | 4 or more rhetoric devices are used correctly and in place. | 3 rhetoric devices are used correctly and in place. | 1-2 rhetoric devices are used correctly and in place. | No rhetoric devices are used. |
| Debate 20% | Controlled and effective concession, refutation and compromise. Excellent rebuttal and defense against the opposing team's objections | There are a few problems with concession, refutation and compromise. missing. Good cross exam and rebuttals, with only minor slip-ups. | Many problems with concession, refutation and compromise. Satisfactory rebuttals, but with some significant problems. | Multiple problems with concession, refutation and compromise. Poor rebuttals, failure to point out problems in the opposing team's position or failure to defend against their attack. |

# Appendix L

# Elective Courses

The following is the list of the elective courses offered to the students. Their actual activation and distribution throughout the semesters depends on factors that vary from year to year, like actual and specific needs of students or of the surrounding industry, availability of professors and of professionals visiting Innopolis University, and so on.

1. Advanced Algorithms for Artificial Intelligence
2. Advanced Software Design
3. Algorithms of Machine Learning
4. Bitcoin and Cryptocurrency Technologies
5. Communication for Startups: From Bootstap to Global Markets
6. Concurrent Object-Oriented Programming with SCOOP
7. Decentralized Applications on Ethereum platform
8. Digital Innovation and Entrepreneurship
9. Distributed systems and middleware: patterns and frameworks
10. Economics of Entrepreneurship
11. Entrepreneurship
12. eSports industry: marketing, economy and game design
13. Ethics and IT
14. Fearless Ideas
15. Formal Software Development of Android Apps
16. Front-end web development
17. Functional Programming and Scala Language
18. Fundamentals of Information Security
19. Human-Computer Interaction Design
20. Industrial C++

21. Industrial programming in Java
22. Industrial Software Testing
23. Innovative Agile Software Development Methodology for High Reliability & Mission Critical Applications
24. International Business: Legal Essentials
25. International Trade: Procedures and Regulations
26. Introduction to Career Development
27. Introduction to Communication
28. Introduction to Convex Optimization
29. Introduction to IT Entrepreneurship
30. Lean Startup Methodology
31. Modern C++: New Language Concepts, Features and Mechanisms
32. Natural Language Processing and Machine Learning
33. Negotiating Processes for the Promotion of Cooperation and Human Rights in International Regimes History
34. Pattern Oriented Design
35. Philosophy of Information
36. Practical Artificial Intelligence
37. Procedural Content Generation for Games
38. Product design
39. Programming in Haskell
40. Reverse Engineering
41. Software Project Management
42. The Principles and Practice of Import/Export
43. User Experience and User Interface Design Fundamentals
44. Venture Capital Hacks: From Zero to Negotiating an Investment Deal
45. Windows Kernel: Architecture and Drivers

## L.1 Advanced Algorithms for Artificial Intelligence

The role of this course is to allow for students to produce a problem worthy of an AI solution and drive the process of solution. The course provides an opportunity for participants to: Display a working understanding of the no-free lunch theory and be able to apply it using a statistical argument; Understand and apply a number of emerging and existing algorithms in AI; Students develop a working project using an element of AI and see it though all steps of the progress from problem description to final presentation and display skills at independent research, finding, reading, and writing technical articles, and display presentation skills

## L.2 Advanced Software Design

This course studies the form of software systems as the product of social design activities. Every software system has an architecture, in the same sense that any building has a form that is the consequence of its function and the needs and social organization of its stakeholders. During the course students presents the most important extra-functional properties of large software systems and study how their structure and components can be reused.

## L.3 Algorithms of Machine Learning

This course teaches students machine learning - the field of study about algorithms that automatically adapt to the analyzed data. This field of study is widely used and actively developing due to growing volumes of collected data and increasing computational resources. During the course students will learn how to make predictive models (classification, regression), how to reduce dimensionality of the data (using feature selection and feature extraction), how to group data into logical categories (clustering), and how to make recommendation systems (collaborative filtering). All these tasks are very common in data rich industries such as web search, telecommunications, banking, marketing and many others. The most famous and widely used algorithms suited to solve these problems are presented. For each algorithm its mathematical model, data assumptions, advantages and disadvantages as well as connections with other algorithms are analyzed to provide an in-depth and critical understanding of the subject. Much attention is given to developing practical skills during the course. Students are asked to apply studied algorithms to real datasets for solving practical problems. Machine learning algorithms are applied using python programming language and its scientific extensions (scikit-learn, numpy, scipy, pandas, matplotlib, seaborn), which are briefly taught during the course.

## L.4 Bitcoin and Cryptocurrency Technologies

The course provides a comprehensive introduction to the revolutionary new technologies of digital currency. How do Bitcoin and its block chain actually work? How secure are your bitcoins? How anonymous are their users? These are some of the main questions this course answers. The course gives the conceptual and practical foundations you need to engineer secure software that interacts with the Bitcoin network as well as to integrate ideas from Bitcoin into your own projects.

## L.5 Communication for Startups: From Bootstrap to Global Markets

This is an intensive course on designing and implementing communication strategies for your current or future startup. During the course, students focus on 1) how to craft a story about your start-up based upon the real needs and problems that it solves, 2) how to implement that story through an investor pitch deck, 3) how to maximize design, websites and apps to reach your audience, 3) tools for content marketing and recruitment of users/clients, 4) how to analyze and improve communications channels, 5) how to work with journalists and the press, 6) how to utilize events to communicate with key audiences, 7) how to organize user feedback sessions, 8) strategic communication for business development opportunities, and 9) crafting full communications plans.

## L.6 Concurrent Object-Oriented Programming with SCOOP

Concurrent programming is more necessary than ever, but remains as hard as ever. Object-oriented mechanisms can in principle facilitate the task, but many concurrent frameworks do not take advantage of the abstraction and correctness guarantees offered by object technology. In this course students analyze the challenges of modern concurrent programming and review the major approaches available, then concentrate on the SCOOP model of computation, which holds promises of reliability and provability. The course has both a conceptual part and a hands-on component, where students will produce a concurrent Object-Oriented application such as a game. All concepts are defined, but it helps if students are already familiar with the fundamentals of concurrent programming. A reading list is provided in advance of the course so that interested students can brush up their knowledge of the field.

## L.7 Decentralized Applications on Ethereum platform

During the course, students learn the fundamental principles and techniques that can be applied to design and develop decentralized applications. This course covers basics of Ethereum platform: clients (Parity, go-ethereum), smart contracts languages (Solidity, Serpent, LLL, Viper), EVM (Ethereum Virtual Machine) and assembly opcodes, popular development frameworks (Truffle, Embark, DappHub), Javascript interface (Web3), Tooling (Remix, TestRPC). During the course, students cover security practices and optimizations and develop several decentralized applications, including ERC20 token application, commonly used in crowdfunding. The course includes graded homeworks.

## L.8 Digital Innovation and Entrepreneurship

This course examines the theory and practice of promoting and managing digital innovation in start-ups and existing firms. It explores successful frameworks, strategies, funding techniques, business models, risks, and barriers for introducing digital innovation. Topics include business model innovation, design-driven innovation, strategy, process improvement, performance measurement.

## L.9 Distributed systems and middleware: patterns and frameworks

The course presents the main technologies to develop distributed systems. The concept of distributed system is introduced first, then the concept of middleware technology. The fundamental principles of distributed systems are explained. This introductory material is followed by a description of the various classes of middleware systems, including those for implementing service-based applications. At the end of the course the students are able to choose the best kind of middleware for a given distributed problem and able to apply the various middleware systems studied to solve practical case studies.

## L.10 Economics of Entrepreneurship

The course provides insights into how economic conditions shape entrepreneurship and vice versa. The course starts with fundamental microeconomics (market outcomes, principal-agency, transaction costs, game theory) and then asks how this informs entrepreneurial outcomes like selection into entrepreneurship through occupational choice theory and firm and market dynamics. The course continues by discussing empirical strategies to assess entrepreneurial outcomes. After that, students continue by discussing the connection between financial structures, agency and entrepreneurial choices. Lastly, students are informed about macroeconomic outcomes of entrepreneurship and its effect on long run economic development. The latter also on the context of optimal entrepreneurial policies.

## L.11 Entrepreneurship

The fourth industrial revolution of an IT based society creates many entrepreneurial opportunities. In addition, complex global challenges like internet security, sustainable business, energy supply, agglomeration, affordable health care require entrepreneurial solutions. Therefore, the study of entrepreneurship is on the rise across the globe and many high-skilled students start ambitious ventures to take advantage of these global opportunities. In addition, most larger firms change their business practices to foster intrapreneurship: the entrepreneurial behaviour of their employees. This course offers a toolbox for both start-up entrepreneurship and corporate entrepreneurship from an international perspective. It places entrepreneurship and innovation in the context of the regional entrepreneurial ecosystem and policy domain, and in particular it addresses international entrepreneurship as a challenge for many young firms in emerging markets.

## L.12 eSports industry: marketing, economy and game design

That is an introductory course to the eSports and the role it plays in the game development and game market. That course lets students study how eSports games are designed, how they retain clients, what's missing on the market today and what is the unit economy of eSports games. In the meantime, eSports course lets students discover what impact video games have on motor and cognitive skills and what is the difference between those games. The goal of the course is to teach students on how to work in eSports industry, create eSports games, balance game strategies and retain clients. Students learn about the games and genres, synthesize information, and explore opportunities in big data related to gamer and fan behaviors in order to develop strategies for effectively targeting Millennials.

## L.13 Ethics and IT

The course explores the ethical and social aspects and problems related to IT and possibilities for resolving, diminishing or preventing these problems. The course teaches students to argue independently about moral, social, and epistemic values and issues that they will encounter in their professional practice. To analyse relevant aspects and concepts, including professional codes, philosophical ethics, and ethical aspects of technological risks. After completing the course, students have basic knowledge of the relevant issues within applied ethics of technology. Students taking the course develop the following academic skills: reading philosophical texts, critical analysis, discussion, and essay writing.

## L.14 Fearless Ideas

The Fearless Ideas Course is Creativity and Innovation Camp. The course teaches students to develop clear business, product or service ideas in co-operation with both students and an expert. During this course the students have the possibility not only to acquire new skills but use these skills for producing innovative business/

product/service concepts.

## L.15 Formal Software Development of Android Apps

The quality of a large extent of today's software systems depends on the skills and experience of Software Engineers rather than on the use of unambiguous technologies. Many practitioners working at software companies are starting to use formal software development alternatives to build software that ensure reliable and correct behavior of applications. These alternatives are based on the use of mathematical models to guide software development. This course covers conceptual and practical underpinnings for formal software development. The course is project base: students use formal methods techniques to develop the core functionality of an Android app.

## L.16 Functional Programming and Scala Language

Functional programming (FP) paradigm becomes one of the mainstream means nowadays. FP is used not only for experimental or academic research, but also in many industrial applications. Most of the popular languages (C++, C#, Java) include features supporting functional programming. Among them, the Scala programming language is perhaps one of the most interesting and promising examples that provide comprehensive set of functional features smoothly integrated with conventional object-oriented paradigm. This course is an introduction to the FP paradigm based on the Scala programming language. During the course, students learn the basic principles and techniques related to the FP approach in design and implementation of the modern software systems. In addition, the students take part in real software projects provided by Innopolis industrial partners (eg, Tinkoff company).

## L.17 Fundamentals of Information Security

This is an introductory course to the field of information security. During this course, students demonstrate understanding of the fundamental principles and techniques that underpin the field of information security engineering from both technical and management point of view. The course covers many fundamental concepts related to information security such as security requirements (confidentiality, integrity, availability) cryptographic techniques, means of authentication, access control policies, risk assessment, information security policies, human resource security, physical security, operations and communications security and compliance. Upon completion of this course, students acquire the necessary understanding and critical thinking for assessing cyber threats based on established risk-assessment methodologies and being in position to contribute to the implementation of an Information Security Management System (ISMS) within an enterprise or organization.

## L.18 Human-Computer Interaction Design

The goal of the course is to teach how to design human-computer interaction (HCI) - whether its a mobile app, a web client, an Arduino-based IoT application, a robot or, as students do in the project, an artificial intelligence. The course focus on the role that users play when designing technology for them and with them. How can students understand what people need and how can students get inspired by what people dream about? How can students translate those needs and dreams into novel technology concepts and improve existing products and services? How can students design user experience? In this class, students understand design not only as a means to create beautiful surfaces but also as an approach that helps students create solutions that people value and enjoy. This class includes project work, in which is applied everything that has been learned to research and solve real-world problems using HCI design methods. Specifically, students address the topic of the student design challenge of CHI 2018, the world's leading research conference on HCI. In addition to what is defined by the challenge, course focus on solutions that use the power of artificial intelligence.

## L.19 Industrial programming in Java

This course is a complete, detailed analysis of the Java programming language, the principles of the JVM, and the use of the basic packages included in the basic delivery of JDK. In the process of training, they train the skills of designing and implementing Java programs, observing the principles of extensibility, performance and fault tolerance. The course allows students to acquire skills in building business applications, messaging systems, and composite applications. Also, the adoption of design decisions based on business requirements, testing of business applications is being worked out.

## L.20 Industrial Software Testing

This is a basic software testing course. The course introduces to the basic approaches to software quality assurance. Students learn in which cases it is necessary to apply a certain type of testing, which levels of testing exist, and to which one should be given maximum attention. In practical part students create test models for software quality determination, conduct manual and automated tests, receive the admission on the software product commercial launch.

## L.21 Innovative Agile Software Development Methodology for High Reliability and Mission Critical Applications

The course is intended for practicing software engineers and software development process managers. Although the specific mention is made to commercial methodologies the students develop their own approach to “agile” for mission critical applications. The lectures present the students with the major issues related to the practical implementation of a development process for high reliability software applications. A particular focus is put on developing a “customer point of view” on the major characteristics of the software product. The 12 (interactive) lectures introduce the main problems connected to the software development life-cycle which the students experiment in the 10 connected exercises (tutored). The course activity helps students to understand mission critical environments and connected risks. The ability to understand complex requirements are tested through the analysis and design of conceptual applications such as Command and Control Systems. Innovative Agile techniques are used to build autonomous team procedures designed by the students.

## L.22 International Business: Legal Essentials

This is a highly practical course in International Business Law. During this course the students have the opportunity to travel the path of a small or medium-size enterprise from set-up/start-up, to launching and growing the business and exiting. The students not only get an insight about the legal issues that come along this way, but also are provided with some hints on how to overcome these legal challenges.

## L.23 International Trade: Procedures and Regulations

This is a highly practical course in International Trade Law. The course covers all of the major concepts of import/ export such as International Sales Contracts, Incoterms® 2010, International Commercial Documents, International Methods of Payment, International Insurance, Transportation, and Customs Clearance. Despite the fact that all of the relevant issues tend to be legal/ technical in nature, they are delivered in plain English.

## L.24 Introduction to Career Development

The course provides students with an opportunity to learn and develop the necessary skills to engage in life/career planning. During the course students learn resume writing and formatting tips, how to go through an interview, and cover job search strategies. In addition, students analyze the main labor market development trends and learn how to develop personal Career Passport.

## L.25 Introduction to Communication

This is an introductory course in communication. During the course, students learn the practical and theoretical foundations of communication theory. Throughout the semester, increasing attention is given to negotiation skills and cross-cultural communication. The aim of the course is to develop negotiation skills experientially. Considerable emphasis is placed on negotiation exercises and role-playing in class, followed by group discussion, lecture and individual analysis.

## L.26 Introduction to Convex Optimization

This is all-covering review course on "typical" optimization methods. During the course, students learn fundamental principles and algorithms of numerical optimization. The principles cover wide range of optimization methods, from basic (gradient-like) to advanced ones. Students learn how to choose appropriate method depending on given problem. The course also provide problem solving and programming experience.

## L.27 Introduction to IT Entrepreneurship

- **Course name:** Introduction to IT Entrepreneurship
- **Course number:** N/A
- **Subject area:**

### L.27.1 Course Characteristics

### L.27.2 What subject area does your course (discipline) belong to?

#### L.27.2.1 Key concepts of the class

- Entrepreneurship
- Startup Creation

#### L.27.2.2 What is the purpose of this course?

The purpose of the Introduction to IT Entrepreneurship program is to enable the course participants to study the fundamentals of entrepreneurship and business operations in context of the Russian and international environment and to study various aspects of the process of new IT business and startup founding. The course will give advanced competences in innovative IT idea commercialization especially in marketing and financial aspects and in startup creation

#### L.27.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize

- the main business-models and types of IT companies
- the main principles of the innovative IT project commercialization
- the basic knowledge of the principles of IT entrepreneurship
- the art of business strategy development for an IT startup

**- What should a student be able to understand at the end of the course?**

- how to create a business-model and financial spreadsheet of an IT project.
- fundamental principles of entrepreneurship and business law and skills of use it in practice
- business planning, marketing research and financial modeling for an IT
- customer development and designing the product or service
- presenting skills for the investment attraction

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- make an investment presentation for an IT seed project
- build a financial model in Excel
- make a marketing research for an IT product or service
- make a customer development field research

#### L.27.2.4 Course evaluation

**Late Submission Policy**

none

Table L.1: Course grade breakdown

| Type | Default Points | Proposed Points |
|---|---|---|
| Labs/seminar classes | 40 | 40 |
| Interim performance assessment | 0 | 0 |
| Exams | 60 | 60 |

**Cooperation Policy and Quotations**

We encourage vigorous discussion and cooperation in this class. You should feel free to discuss any aspects of the class with any classmates. However, we insist that any written material that is not specifically designated as a Team Deliverable be done by you alone. This includes answers to reading questions, individual reports associated with assignments, and labs. We also insist that if you include verbatim text from any source, you clearly indicate it using standard conventions of quotation or indentation and a note to indicate the source.

#### L.27.2.5 Grades range

Table L.2: Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

#### L.27.2.6 Resources and reference material

Textbook(s)

- Brealey R.A., Myers S.C. (2003) Principles of Corporate Finance - The McGraw-Hill Companies.
- Damodaran A. Investment Valuation: Tools and Techniques for Determining the Value of Any Asset (2012) Wiley Finance
- Entrepreneurship – Paperback Series, Harvard Business Review

- Gompers Paul. A. and Lerner Josh (2001) The Money of Invention - Harvard Business School Press
- Lerner Josh, Hardymon Felda, and Leamon Ann (2005): A Casebook - Wiley Text Books
- Timmons Jeffry. A., Spinelli Stephen (2007). New Venture Creation. Entrepreneurship for the 21st Century –McGraw-Hill

Reference Materials

- Moore Geoffrey A. (1991) Crossing the Chasm - Harper Collins
- Moore Geoffrey A. (1995) Inside the Tornado - Harper Collins
- Ries Eric (2011). The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Publishing.

Computer Resources

- Students should have laptops with Internet access

## L.27.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table L.3:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | The fundamental principles of IT entrepreneurship | 18 |
| 2 | Product, market, business model of an IT project | 18 |
| 3 | Market advance – the main source of expenses | 18 |
| 4 | Financial plan of an IT startup | 18 |
| 5 | Incorporate IT startup | 18 |
| 6 | Project management in commercialization of innovations | 18 |

### L.27.3.1 Section 1

**Section title:** The fundamental principles of IT entrepreneurship

**Topics covered in this section:**

- The role of an entrepreneur and entrepreneurship in IT economy
- Differences between entrepreneurship and other types of a business management and economic activities
- The main entrepreneur's competences
- The aspects of the entrepreneurship statutory regulation
- How to start a business
- Different types of the enterprise companies: Software Business, Internet services, eFinancial projects,Infobusiness, e-learning.
- Companies in university ecosystem, research institutes, corporations.
- Practical recommendations: which sectors are favorable for creation the IT company in Russia

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | no |
| Reports | no |
| Essays | no |
| Oral polls | no |
| Discussions | no |

**Typical questions for ongoing performance evaluation within this section**

1. Create the idea of your own startup project.
2. Define the key competitive advantage.
3. Define the time framework of the realization of the project.
4. Present the key competitive advantage in three minutes in oral form

**Typical questions for seminar classes (labs) within this section**

1. Create the idea of your own startup project
2. Define the key competitive advantage
3. Define the time framework of the realization

**Test questions for final assessment in this section**

1. How you can describe the key competitive advantage?
2. How the planning of time should be done?
3. What are basic principles of the oral presentation of project idea?

#### L.27.3.2 Section 2

**Section title:** Product, market, business model of an IT project

**Topics covered in this section:**

- Business-model as the base of competitiveness and the innovation startup strategy
- Parameters and metrics of business-model quality
- Business model canvas
- Business ideas generation: how to create the concept suitable for opening of business
- Methods of collective working on a business idea
- Mission of the company. Fast profit or company capitalization growth - what to choose? Building of the business "for the constant dividends" or " for sale
- Business model: initial development and testing
- Problem of business scaling. How to scale an IT business
- Key performance indicators as reflection of the purposes and tasks of a company, leader and certain employee
- The use of “roadmap” technology: what is it, for what purpose it should be created, the steps of its creation
- Steve Blank’s Customer Development concept
- Business-model evaluation, importance of the consumers’ feedback
- How to define the need of the business-model correction

**What forms of evaluation are used to test students’ performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

1. Define the size of the market
2. Create the program of testing the market

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | no |
| Reports | no |
| Essays | no |
| Oral polls | no |
| Discussions | no |

3. Start the testing of the market according to the CusDev approach
4. Present orally the first results of CusDev for your project.

**Test questions for final assessment in this section**

1. What are the principles of market size defining?
2. What are the fundamental characteristics of CusDev approach?
3. What are the peculiarities of test market approach for an IT service?

#### L.27.3.3 Section 3

**Section title:** Market advance – the main source of expenses

**Topics covered in this section:**

- Innovation's life cycle. G. Moore's Approach of "Crossing the Chasm" to a product introduction to the mass market. How to penetrate the Blue Ocean and skip the competitors?
- Chris Anderson's "Long tail" concept - organization of a large number of small diverse sales online
- Instruments of low budget market promotion of an IT product or service
- Outlining of a marketing plan and marketing budget of an IT startup
- Laws of business and investment attraction
- Basic principles of creation of the modern innovative company
- Opening of the independent startup company. Investor attracting and involvement. Methods of communications with investors
- Basics of Investor Relations: an object of the search, methods of a search for the investor,VC and Seed market, public grants in Russia, negotiation with investors, documentary registration of a deal with investors, necessary documents for the

investor,venture and business angel investing principles. How to attract an IT investment in Russia

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | no |
| Reports | no |
| Essays | no |
| Oral polls | no |
| Discussions | no |

**Typical questions for ongoing performance evaluation within this section**

1. Find the seed funds which can finance your project.

**Typical questions for seminar classes (labs) within this section**

1. Find the VC funds which can finance your project
2. Find the grants which can finance your project
3. What could be the possible path of the Exit

**Test questions for final assessment in this section**

1. What is the difference between seed and venture financing?
2. What are basic principles of the exit?
3. What the most important grants in Russia for startup can you mention?

### L.27.3.4 Section 4

**Section title:** Financial plan of an IT startup

**Topics covered in this section:**

- Business planning. The structure for a business plan. The business plan "for investors" and "for everyday life". Differences in approaches
- Drawing up the financial model,forecasting the financial indicators

- P&L, Cash Flow, Balance Sheet, main indicators of investment profitability, discount factors, NPV, IRR
- Calculation of a discount rate. Calculation and justification of the investor's or creditor's benefits
- Presentation for investors, presenting skills
- Standard criteria of the IT project attractiveness for investors
- Intellectual property management, business sale, investor's "exit" from the IT project

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | no |
| Reports | no |
| Essays | no |
| Oral polls | no |
| Discussions | no |

**Typical questions for ongoing performance evaluation within this section**

1. Calculate the CAPEX values for the project.

**Typical questions for seminar classes (labs) within this section**

1. Create the P&L and Cash Flow of the project
2. Define the Terminal Value
3. Calculate the NPV and IRR of the project

**Test questions for final assessment in this section**

1. What are basic principles for P&L and Cash Flow calculation?
2. How the terminal value should be calculated by two methods?
3. What are the basic principles of IRR and NPV calculation?

### L.27.3.5 Section 5

**Section title:** Intracorporate IT startup

**Topics covered in this section:**

- The instruments of realization and financing startup within corporation,key competences
- 'Open innovations' principle, the main methods of the 'Open innovations' principle' realization
- How a corporation can help an internal entrepreneur

Using the methods of the 'Open innovations' in the world's practice:

- Intel. Work with universities and research institutes and corporate venturing
- IBM. From a full cycle of development of a product to participation in production chains
- Siemens. Open policy of creation the development processes results
- HP. Experimental environment of testing and prototyping
- Cisco. External entrepreneurship spin In. Internal Entrepreneurship
- Philips. Encouragement of external creation of knowledge

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | no |
| Reports | no |
| Essays | no |
| Oral polls | no |
| Discussions | no |

**Typical questions for ongoing performance evaluation within this section**

1. Please discuss what international corporation can have an interest in your project.

**Typical questions for seminar classes (labs) within this section**

1. Find the corporations, which can buy your startup
2. Define the value by which they can do so
3. By which structure (may be the intracorporate VF) they can do it

**Test questions for final assessment in this section**

1. What are the types of structures of possible interaction between corporations and startups?
2. What are the basic principles of corporate VC fund functioning?
3. What is the differences between functioning of independent and corporate fund/accelerator?

#### L.27.3.6 Section 6

**Section title:** Project management in commercialization of innovations

**Topics covered in this section:**

- Phases of innovation lifecycle project management: research activity, R&D project, prototype, trial production, production manufacturing
- The Stage-Gate process: project concepts, project business plan, development, testing and confirmation, startup phase

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | no |
| Reports | no |
| Essays | no |
| Oral polls | no |
| Discussions | no |

**Typical questions for ongoing performance evaluation within this section**

1. Create the plan if the pivot is needed.

**Typical questions for seminar classes (labs) within this section**

1. Create the plan for the first 100 days
2. Create the plan for the next 3 months
3. Create the plan if the pivot is needed

**Test questions for final assessment in this section**

1. What is the difference in short and long-term planning?
2. What are the basic principles of short-time budgeting?
3. What are the basic conditions for a successful pivot?

## L.28 Lean Start-up Methodology

This is an introductory course to the Start-up Development Process not only for those who plan to start their own companies but also for those who plan to work as a team lead or product manager in a company or the governmental organization. Starting from basics the course guides students on how to start a project from scratch, develop a customer base, validate hypothesis and scale up. Course introduces students to the world of business models, lean start-up methodology, customer development, key metrics, team management, investor relationships, competitive analysis and tactics to find a blue ocean. One of the main reasons why startups and projects fail is incapability of founders to find the real pain of customers and choose the right market. By completing this course students learn how to provide a solution for the market, how start-up is working and how it's different from the classic business. Acquired skills help to come up with the execution plan for the own future start-up and in the meantime help to commit to the corporate's business as an entrepreneur in house.

## L.29 Modern C++

C++ still remains the most widely used language in industry for implementing big and very big software projects. The language is actively evolving; the C++ Working Group within the ISO is working hard issuing new language standards. The latest editions were C++14 and C++17. Many new important concepts and features were added to the language, and the new standard C++20 is being prepared. This requires developers to keep in sync with the newest updates in the language. Also, some important language features added to C++ even before 2014, are still not in common use because of its relative complexity and the absence of clear explanations (textbooks, papers etc.). The course aims to overcome the gap between the current state of the C++ language and the limited subset of the language that is in actual use by C++ developers. Most important new language features are carefully introduced, discussed, and explained on typical code examples.

## L.30 Natural Language Processing and Machine Learning

This course gives a tour of data-intensive natural language processing (NLP) and machine learning (ML). Students study automated processing of the big data source that is social media, from an artificial intelligence perspective, using statistical machine learning (ML). The course introduces the field, and demonstrate multiple complete practical examples in various real-world tasks. Covering basic structures and theory, the course leaves participants with knowledge of how to match machine learning tool with problem in the context of language processing. The intricacies and nuances involved in social media text are also covered, being a significantly different and more interesting text type than the prior research on newswire. Topics covered include entity recognition, sentiment analysis, discriminative and structured learning, and processing for indexing and retrieval. The machine learning skills required to complete the course are included (i.e. types of learner, representations, and evaluation). The later parts of the course give both theory and practical tools for machine learning.

## L.31 Negotiating Processes for the Promotion of Cooperation and Human Rights in International Regimes History

The course explores notions of international community law, the historical development of the cooperation of international political institutions. Students study jurisdictional sources of the guarantee of international law, the promotion and protection of cooperation and multicultural society through the UN system. The course teaches students to international protection of human rights, hierarchy of human rights and the human rights system from the Geneva Convention up to the present.

## L.32 Pattern Oriented Design

This course explains the issues and means in designing software systems for reuse and extension. Students understand how to leverage the powers of object-orientation embodied in well-known heuristics, principles and patterns in the design and construction of reusable frameworks, packages and components.

## L.33 Philosophy of Information

Information is now a vitally important scientific concept, while changes in information and communication technologies have rapidly altered our personal and working lives. In this course, philosophy is interpreted as the conceptual design of cogent and relevant answers to open questions of a foundational nature. It introduces some key concepts and phenomena related to information. And it seeks to answer some crucial theoretical questions of great philosophical significance prompted by the development of the information society.

## L.34 Practical Artificial Intelligence

- **Course name:** Practical Artificial Intelligence

- **Course number:** N/A

## L.34.1 Course Characteristics

## L.34.2 What subject area does your course (discipline) belong to?

Artificial intelligence

### L.34.2.1 Key concepts of the class

- Practical tools to implement artificial intelligence
- Methods and approaches to solving intellectual tasks

### L.34.2.2 What is the purpose of this course?

The course gives a general overview of the history, theoretical basis and technological stack for what we now call "artificial intelligence" (AI). Today AI is not only a research area, but also a complex set of exact algorithms, technologies, frameworks, software and services, which can be easily integrated in modern software. In this course students will learn the history, major theoretical points, structure of knowledge related to AI. The major goal of the course is to practice contemporary AI technologies and frameworks, including reasoning, natural language processing, computer vision, and machine learning. Working individually and in teams, students will solve a variety of AI problems, both from scratch and using existing solutions.

### L.34.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize

- Ways to present and describe artificial intelligence in software solutions
- The best developed intelligent technologies that perform with near-human quality
- Ideas behind building intelligent systems in reasoning, natural language processing, recommendations, computer vision, speech processing

- Necessary parts to build an intelligent system with computer vision and machine learning technologies
- Factors to decide whether to go for AI or stay in conventional software development approach

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- Major approaches to construct AI solutions
- Popular libraries and APIs to solve AI tasks
- Algorithms to construct intelligent agents from scratch
- Basic methods of image processing and machine learning
- Major approaches to natural language and speech understanding

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Create game playing bots using heuristic approaches
- Implement knowledge databases
- Implement systems with speech processing and generation
- Implement image processing and computer vision solutions
- Perform basic data analysis using machine learning

#### L.34.2.4 Course evaluation

**Table L.4:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 0 |
| Interim performance assessment | 30 | 0 |
| Assessments | 0 | 60 |
| Exams | 50 | 40 |

If necessary, please indicate freely your course's features in terms of students' performance assessment:

Lectures and labs are followed by the home works (14). Home works are covering 60% of the grade. 40% of the grade fall to exam session, which will be in the form of solving practical task, taking 1 hour for best student, 2 hours for average students and 3 hours cut-off deadline.

#### L.34.2.5 Grades range

**Table L.5:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 60-79 |
| C. Satisfactory | 60-74 | 40-59 |
| D. Poor | 0-59 | 0-39 |

#### L.34.2.6 Resources and reference material

Main textbook:

- “Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit,” Steven Bird, Ewan Klein, and Edward Loper. [link]
- Practical Python and OpenCV. [link]

Other reference material:

- Ray Kurzweil. The Singularity Is Near: When Humans Transcend Biology. [link]
- Amazon Polly: Text-To-Speech service. [link]
- Get Started with TensorFlow. [link]
- Natural Language Processing with Python. [link]
- Hands-On Machine Learning with Scikit-Learn and TensorFlow. [link]

### L.34.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table L.6:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction and reasoning | 12 |
| 2 | Text and speech processing | 16 |
| 3 | Images and video processing | 12 |
| 4 | ML for practice. Recommendations | 20 |

### L.34.3.1 Section 1

**Section title:** Introduction and reasoning

**Topics covered in this section:**

- Introduction to artificial intelligence
- History of artificial intelligence
- Considering AI as function
- Reasoning approaches

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Implement `https://en.wikipedia.org/wiki/Nim` game bot
2. Solve simple problems for CLIPS-based expect system completion
3. Run inference for Bayesian network with given observations
4. Discuss article in media about artificial intelligence breakthrough

**Typical questions for seminar classes (labs) within this section**

1. Implement tic-tac-toe bot
2. Solve non-linear equation with symbolic computation package
3. Implement Wolfram Alpha bot
4. Implement decision tree framework

**Test questions for final assessment in this section**

1. Implement a bot to answer general question on math
2. Find local minima and maxima of the function
3. Implement an expert system to detect graph cycles
4. Implement a bot playing simple board game with minimax algorithm

### L.34.3.2 Section 2

**Section title:** Text and speech processing

**Topics covered in this section:**

- Natural language processing (NLP) for syntax analysis
- NLP for semantic analysis
- Speech processing

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Build a syntax tree of a sentence
2. Determine token types for all words in a sentence
3. Recognize text in a file with speech
4. Visualize histogram of a wave file

**Typical questions for seminar classes (labs) within this section**

1. Extract subject and object from a sentence
2. Visualize text similarity in a dataset with doc2vec and t-SNE
3. State a problem for symbolic algebra freamework in natural language
4. Implement a command with a voice message

5. Implement simple programming language
6. Detect a chord by a wave file

**Test questions for final assessment in this section**

1. Implement programming language for a given description
2. Implement a bot, that pronounces a fact on a given topic
3. Implement comparison of two texts with respect to syntactic complexity

### L.34.3.3 Section 3

**Section title:** Images and video processing

**Topics covered in this section:**

- Image processing with OpenCV
- Models for image processing
- Machine learning for image understanding

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Count simple object in a picture
2. Label objects on a picture
3. Filter out noisy image
4. Find contours of cookies on the image

**Typical questions for seminar classes (labs) within this section**

1. Build an eye detector
2. Use SIFT descriptors to detect images using bag of words representation

3. Subtract and measure fishes on a contract background
4. Track labelled objects in a video

**Test questions for final assessment in this section**

1. Recognize handwritten text
2. Convert chessboard image into notation
3. Measure a distance the cat cover in the video
4. How many balls are there on the image?

#### L.34.3.4 Section 4

**Section title:** Machine learning for practice. Recommendations

**Topics covered in this section:**

- Machine learning as a framework
- Recommendation
- Classification practicum
- Clustering practicum

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Run PCA using SVD for a given matrix
2. Remove unnecessary features
3. Split a dataset for a k-fold cross-validation
4. Perform a grid search of hyper-parameters

**Typical questions for seminar classes (labs) within this section**

1. Find explainable predictors for a real data
2. Separate cats from dogs in a dataset
3. Train and estimate a model using k-fold cross-validation
4. Detect how similar are the interests of 2 users

**Test questions for final assessment in this section**

1. Implement person classifier by voice
2. Implement human classifier by face
3. Implement recommender system for cold start
4. Implement recommender system for a provided dataset

## L.35 Procedural Content Generation for Games

Procedural Content Generation (PCG) allows for parameter models of designs to create content with minimal user interventions, or to provide a human with an overview of possible designs. Techniques involving generative methods, search based algorithms, and evolution, give the designer control over large design spaces. These techniques are at the forefront in novel tools for game design and assets: expanding replay-ability, reducing costs and entry points for asset creation, and allowing for personalized play experiences. These methods have also found their way into other industries, such as, fashion, architecture, and industrial design. This course aims to provide a project based approach to learning these techniques. Students finish the course with a portfolio of designs generated by PCG methods, understand how to use parameter models for designs and evaluate the designs using quantitative measures, and have a foundation in computational creativity.

## L.36 Product design

This is an advanced course for students who took UX and HCI related courses or have relevant experience and want to continue studying and advance in the field. While still maintaining the user centered focus on software interface design this course offers new materials on visual design, branding, analytics, problem solving and design thinking. Product design course strives to give a holistic view on software product design, its promotion and development.

## L.37 Programming in Haskell

- **Course name:** Programming in Haskell
- **Course number:** XYZ
- **Knowledge area:** Software Engineering, Software design patterns

### L.37.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 3rd and 4th years of BS
- **Semester of instruction:** Any
- **No. of Credits:** 5 ECTS
- **Total workload on average:** 12 hours per week including 8 hours of self-study
- **Frontal lecture hours:** 2 per week.
- **Frontal tutorial hours:** 0.
- **Lab hours:** 2 per week.
- **Individual lab hours:** 0.
- **Frequency:** weekly throughout the semester.
- **Grading mode:** letters: A, B, C, D.

### L.37.2 Course outline

Haskell is an advanced purely functional programming language with strong static type system and elegant mathematical background. It is being increasingly used by organizations like Facebook and Kaspersky Lab along with various companies in finance, blockchain, advertising, data science, scientific computing and more. In this course we will explore practical functional programming in Haskell. In the first part of this course students will learn the necessary Haskell basics to easily write practical programs, covering higher-order functions, parametric polymorphism, lazy evaluation, algebraic data types and type classes. The second part of the course will introduce some of the abstractions commonly used in functional software design, such as monoids, functors and monads. The last few weeks will introduce students to several more advanced topics. Evaluation will be based on class participation, programming assignments, and an open-ended final project completed individually or in small teams.

### L.37.3 Expected learning outcomes

- understand benefits and tradeoffs of purely functional programming,
- understand, verify and test properties of functional programs,
- reason about programs using equational reasoning,
- closely model various domains with algebraic data types,
- become familiar with abstractions representing some common software design patterns,
- create practical projects in Haskell.

### L.37.4 Expected acquired core competences

- programming in purely functional style,
- modelling domains using algebraic data types,
- using and applying higher-order functions, parametric polymorphism, lazy evaluation and type classes for program structure;
- recognising and using `Semigroup` and `Monoid` abstration in some domains;
- using `Functor`, `Applicative` and `Monad` type classes for handling effects in purely functional code;

### L.37.5 Prerequisites

It is recommended to pass **Data Structures and Algorithms** as well as **Discrete math / Logic** courses before Programming in Haskell to better understand some of the examples and design decisions. Experience from **Software Project** course will be helpful to draw analogies and analyze differences in OO and FP software design.

Some students might benefit from **Introduction to Artificial Intelligence** and **Data Modeling and Databases** courses for stage II of their final projects.

### L.37.6 Textbook

- Hutton, G. (2016). *Programming in Haskell.* Cambridge: Cambridge University Press.

### L.37.7 Reference material

- Learn You a Haskell for Great Good! — `http://learnyouahaskell.com/`
- Standard library documentation — `http://hackage.haskell.org/package/base`
- Hackage (Haskell repository of packages) — `http://hackage.haskell.org/`
- Hoogle (Haskell code search engine) — `http://www.haskell.org/hoogle/` or`https://hoogle.haskell.org`
- Hayoo (another Haskell code search engine) — `http://holumbus.fh-wedel.de/hayoo/hayoo.html`
- Functional Pearls — `https://wiki.haskell.org/Research_papers/Functional_pearls`
- Haskell Weekly — `https://haskellweekly.news/`

### L.37.8 Required computer resources

Students should have laptops with the latest version of Haskell Platform installed.

### L.37.9 Evaluation

- Weekly homework assignments (30%)
- Final Project Stage I (20%)
- Final Project Stage II (40%)
- Class and lab participation (10%)

## L.38 Reverse Engineering

This course covers various areas of system programming, white hacking, reverse engineering, disassembling/decompilation and advanced C topics like exception handling. Another point is to teach how does programs looks at binary level of machine code. Besides lectures, this course includes a substantial part of practice. Students get task description and starting code at labs, but require completing assignments at home. This course requires some amount of time for self-study and homework.

## L.39 Software Project Management

According to Project Management Institute (www.pmi.org) project management is "the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements". As such this module not only builds the required project management body of knowledge but it also presents tools and techniques needed for successful project management. Knowledge areas required by a project manager for managing successfully project such as, project requirements management, change management, cost management, time management, quality management, risk management, communication management, etc. Students are presented in detail, along with a number of tools for managing the respective processes. Further, emphasis is given to peculiarities of software projects, covering topics such as software life-cycle models, effective estimation of software effort and size, software quality metrics and team management. Finally, since this module is targeted to computer science and engineering students an attempt is made to establish the link between project management and organization's strategy.

## L.40 The Principles and Practice of Import/Export

Some years back, international trade was dominated by very large companies. From then onwards, much has changed on the world economic scene. Today, when international trade is growing to unprecedented levels, more and more businesses (mostly SME) are taking advantage of the numerous opportunities and benefits to be gained from trading in a global market place. This course demystifies international trade and introduces you to key principles and practices of import/export operations. Students go on to study the subject in more depth, acquire some basic knowledge and an understanding of the particular topic.

## L.41 User Experience and User Interface Design Fundamentals

The course focuses on what software engineers, architects, product owners and project managers need to do to create usable designs, explains the importance of user experience and provides methods and instruments for integrating UX in the project lifecycle. This course is useful for beginners and people who have some experience in software engineering.

## L.42 Venture Capital Hacks: From Zero to Negotiating an Investment Deal

This is an intensive course on investment strategies for student's current or future start-up. In the first series of the course students delve into the key aspects of building successful start-ups that get funded by global Venture Capitalist which includes 1) how to identify compelling growth markets 2) how to find the right product-market fit through the customer development approach and 3) how to pitch a start-up to investors. Throughout the course students learn what investors look for in start-ups, how to think like an investor and speak their 'language'. In the second series of the course students further delve into negotiations strategies in order to get favorable deal terms with Venture Capitalist which includes 1) how to identify the right investor and negotiate a term-sheet 2) how to value a start-up 3) how to negotiate the final investment agreement with founders friendly terms.

## L.43 Windows Kernel: Architecture and Drivers

This course covers Windows kernel architecture topics required for kernel driver developer in file-systems area. During the course, students learn Windows kernel architectural concepts and principles and apply this knowledge / train skills in drivers' development. Besides lectures, this course includes a substantial part of practice. Students get task description and starting code at labs, but require completing assignments at home. This course requires some amount of time for self-study and homework.

## L.44 Global Business, International Marketing and International Trade Law in IT industry

- **Course name:** Global Business, International Marketing and International Trade Law in IT industry
- **Course number:** N/A
- **Subject area:**

### L.44.1 Course Characteristics

### L.44.2 What subject area does your course (discipline) belong to?

Global Business, International, World Economy and Trade, Globalization, Internationalization and Multi-nationalization, IB Operations, Localization, International Trade, Procedures and Regulations, Culture and Intercultural Communication, Global Marketing, International Communication Decision, International Contracts, International Terms of Trade, International Terms of Payment, International Commercial Documents, International Air Transportation, International Land and Multi modal Transportation.

#### L.44.2.1 Key concepts of the class:

- the development of Global Business, International Marketing, and International Trade activities from the determination of objectives and methods of the organisation through the execution of research, advertising, distribution, and production activities

#### L.44.2.2 What is the purpose of this course?

Course competency, a student should have the ability to analyze, discuss, describe, and demonstrate the Global Business processes and strategies that companies utilise when "move" their products to foreign markets.

#### L.44.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize

- what differentiates international from domestic business
- the major functional pieces of an International Sales Transaction
- the various legislative rules associated with international sales transactions and the movement of freight internationally
- basic features of selected European and world national cultures

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- the context in which international business takes place
- current situation in a world economy, in a world trade
- multi nationalization of companies
- the major functional pieces of an International Sales Transaction
- the use of various documents used in international transactions
- the various modes of transportation
- various legislative rules associated with international sales transactions and the movement of freight internationally

- the International Marine Cargo Insurance
- the customs clearance process
- the concept of culture and its nature and the concept of cultural dimensions
- the EU common commercial policy and a development of its relationship with Russian Federation

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- evaluate issues in the global marketing mix decisions of product, price, place and promotion
- summarize an obtained knowledge in own presentation (internationalization process of a chosen MNC)
- to use most appropriate Incoterms® rule for the shipment of a product
- create an import/export plan in a professional way
- design an import/export plan taking into account the demand and supply factors, including the legislative framework and logistics, in the context of the product and market area as well as a company's resources.

#### L.44.2.4 Course evaluation

**Table L.7:** Course grade breakdown

| Type | Default Points | Proposed Points |
|---|---|---|
| Interaction | | 25 |
| Groupwork | | 20 |
| Exam | | 55 |

**Late Submission Policy**

none

**Cooperation Policy and Quotations**

We encourage vigorous discussion and cooperation in this class. You should feel free to discuss any aspects of the class with any classmates. However, we insist that any written material that is not specifically designated as a Team Deliverable be done by you alone. This includes answers to reading questions, individual reports associated with assignments, and labs. We also insist that if you include verbatim text from any source, you clearly indicate it using standard conventions of quotation or indentation and a note to indicate the source.

#### L.44.2.5 Grades range

**Table L.8:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

#### L.44.2.6 Resources and reference material

- ALBAUM G, DUERR E, Edwin Duerr, 2008/2012/2017. International Marketing and Export Management. Financial Times/ Prentice Hall ISBN: 978-1292016924 ISBN: 978-0273743880 ISBN: 978-8131791189
- DANIELS, J. D., RADEBAUGH, L. H., SULLIVAN, D. P. International Business: Environments and Operations.. ISBN 978-01-3266-.
- KRUGMAN, P., OBSTFELD, M., MELITZ, M. International Economics. Theory and Policy. ISBN 978-01-3342-
- DANIELS, J. D., RADEBAUGH, L. H., SULLIVAN, D. P. International Business: Environments and Operations.. ISBN 978-01-3266-.
- KRUGMAN, P., OBSTFELD, M., MELITZ, M. International Economics. Theory and Policy. ISBN 978-01-3342-

### L.44.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table L.9:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Global Business | |
| 2 | International Trade Law | |
| 3 | International Marketing | |
| 4 | World Economy and Trade | |
| 5 | Culture and Intercultural Communication | |

### L.44.3.1 Section 1

**Section title:** Global Business

**Topics covered in this section:**

- Introduction to Global Business. Factors in IB Operations. Globalization and Localization. Competitive advantage. Communication Decision. Financial Decision, Internalization Logic. Global Marketing and the Digital Revolution
- International Marketing, Regional operations, Cultural aspects
- 6 Brand and Product Decisions in Global Marketing, Importance of "country of origin"
- Strategic Elements of Competitive Advantage
- Personal Development

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | yes |
| Reports | yes |
| Essays | yes |
| Oral polls | yes |
| Discussions | yes |

**Typical questions for ongoing performance evaluation within this section**

1. What are the risks of entering an international market?
2. How can a company obtain the expertise needed to produce and market its products in, for example the EU?
3. Discuss the general functions involved in international cash management

**Typical questions for seminar classes (labs) within this section**

1. Organizations often find that they need to develop new skills and abilities in order to expand their operations internationally. Consider the differences between operating in one country and operating in many countries. Write an essay describing the skills and abilities that an organization would need to develop in order to compete in the global marketplace. Explain why those skills and abilities are especially important in an international setting. Provide examples to support your point of view.
2. Suppose there are two goods being produced in the economy: robots and wheat. Robots are a capital intensive good and wheat is a labor intensive good. Suppose that there are two countries: France and Brazil. France is capital abundant, and Brazil is labor abundant. Using two graphs (one per country) draw the international trade equilibrium for this two-country world economy based on the Heckscher-Ohlin model. Be sure to include (1) the production possibility frontiers, (2) the relative price line, (3) indifference curves and (4) trade triangles. Completely label both graphs for full credit.

**Test questions for final assessment in this section**

1. What are the acceptable methods of accounting for business operations in a foreign country?
2. Choose a publicly traded company that operates internationally and identify impact that the foreign operations have on the financial statements. Explain.

### L.44.3.2 Section 2

**Section title:** International Trade Law

**Topics covered in this section:**

- International Contracts
- International Terms of Trade; International Terms of Payment; International Commercial Documents; International Ocean Transportation
- International Air Transportation. International Land and Multimodal Transportation

- International Insurance. Customs Clearance
- Personal Development

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | yes |
| Reports | yes |
| Essays | yes |
| Oral polls | yes |
| Discussions | yes |

**Typical questions for ongoing performance evaluation within this section**

1. What are oral contracts? Are they ever enforceable?
2. What are implied contracts?
3. What are the basic functions of commercial invoices in international trade?.

**Typical questions for seminar classes (labs) within this section**

1. What are the strategic advantages for an exporter of having good document preparation.
2. What is the impact on the shift from Panamax to post-Panamax on the Panama Canal and shipping in general?

**Test questions for final assessment in this section**

1. Identify basic international regulations regarding air transport.
2. Do a report on containerization, including its history, advantages, problems, and future.

#### L.44.3.3 Section 3

**Section title:** International Marketing

**Topics covered in this section:**

- Consultations
- Personal Development
- Group Development

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | yes |
| Reports | yes |
| Essays | yes |
| Oral polls | yes |
| Discussions | yes |

**Typical questions for ongoing performance evaluation within this section**

1. List some forms of political risk other than a takeover of a subsidiary by the host government, and briefly elaborate on how each factor can affect the risk to the MNC.
2. Identify common financial factors for an MNC to consider when assessing country risk.
3. Briefly elaborate on how each factor can affect the risk to the MNC.

**Typical questions for seminar classes (labs) within this section**

1. Why do you think MNCs continuously assess possible forms of multinational restructuring, such as foreign acquisitions or downsizing of a foreign subsidiary? (case study)
2. Why did Tesco's initial internationals expansion strategy focus on developing nations? (case study)

**Test questions for final assessment in this section**

1. List some forms of political risk other than a takeover of a subsidiary by the host government, and briefly elaborate on how each factor can affect the risk to the MNC.
2. Identify common financial factors for an MNC to consider when assessing country risk.
3. Briefly elaborate on how each factor can affect the risk to the MNC.

### L.44.3.4 Section 4

**Section title:** World Economy and Trade

**Topics covered in this section:**

- International trade environment. Specifics of international trade in individual sectors (A, I, S). Integrations. EU Common Commercial Policy
- International movement of production factors - capital, human resources. FDI.
- Internalization of companies – circumstances. Models and areas of internationalization. MNC structures and management

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | yes |
| Reports | yes |
| Essays | yes |
| Oral polls | yes |
| Public talk | yes |

**Typical questions for ongoing performance evaluation within this section**

1. What are the ultimate impacts of temporary expansionary monetary policy under fixed exchange rates on Y, i, E and the TB? Briefly explain.
2. What are the ultimate impacts of temporary fiscal contraction under floating exchange rates on Y, i, E, and the TB? Briefly explain.
3. Consider the open economy model. Suppose that U.S. investors decide to invest more in Canada. What happens to the real interest rate, the real exchange rate, the net capital outflow, and the net exports in Canada and in the U.S.?

**Typical questions for seminar classes (labs) within this section**

1. Do some research on the Russian Federation. After doing your research, please consider recent events and the prevailing political and economic conditions in the Russian Federation. As a manager who has been considering investment there, how do you assess the political and economic risks at this time? What should be your company's response to this environment? Explain your answer
2. The government want to drive the price of soybeans above the equilibrium price, p1 to p2. it offers growers a lump sum payment of x to reduce their output from Q1 (the equilibrium level) to Q2, which is the quantity demanded by consumers

at p2. Use a figure to show how large x must be (an area in the figure) for growers to reduce output to this level. What are the effects of this program on consumers, farmers, and total welfare?

**Test questions for final assessment in this section**

1. Provide graduate and advance answer for the following question: - Discuss and comment on the proposition that in the final analysis the theory of growth is simply about capital theory. Use any particular growth theory to support your comment and discussion.
2. Provide graduate and advance answer for the following question: -The stock market is reacting to the value of the dollar vs the Yen. Discuss the current value of the dollar to the Yen and explain how a strong dollar and weak dollar affects the economy.

## L.45 Tech Startup Design

- **Course name:** Tech Startup Design
- **Course number:** N/A

### L.45.1 Course Characteristics

### L.45.2 What subject area does your course (discipline) belong to?

#### L.45.2.1 Key concepts of the class

- design-fiction
- design-thinking
- business-design
- Agile development (SCRUM)
- product management

#### L.45.2.2 What is the purpose of this course?

The start-up world is rapidly developing nowadays, more and more people are considering the opportunity to set up their own business. But how can we turn the idea into the working business? The purpose of the course is to walk students through the concrete steps that are necessary for an entrepreneur to develop a tech product and build a solid business around that tech product.

#### L.45.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize:

- design-fiction tools to design the future
- design-thinking tools to design the prototype of the product
- approaches to designing and testing a business model through the experiments
- frameworks of agile development
- product management frameworks
- pitching presentation tools

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples):

- value of design-fiction to predict the future and define tech trends
- concrete steps of the design-thinking methodology
- concrete steps of the business design (business model, hypothesis formulation/testing and minimum-viable product creation)
- SCRUM roles, ceremonies and artefacts
- product manager's role and its specifics
- specifics of pitch presentation for investors

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply:

- design-fiction methods to define the vision and mission of the company
- design-thinking methodology (empathy developing tools, ideation tools, prototyping and testing tools) to develop user-oriented products
- business model generation approaches to develop innovative business models and test their assumptions, develop the minimum viable product (MVP)
- agile development (SCRUM) tools
- develop products using the agile method SCRUM
- product management tools to manage the product through its life-cycle
- presentation skills for pitching for investors

**What should a student be able to analyse at the end of the course?**

By the end of the course, the students should be able to analyse:

- scientific literature about emerging technologies,
- customer behavioral patterns and customer feedback,
- tech product functionality,
- tech product lifecycle.

**What should a student be able to evaluate at the end of the course?**

By the end of the course, the students should be able to evaluate:

- emerging technologies,
- customer behavior and context of the product use,
- tech prototypes,
- MVP's functionality,
- business models, cost structure and revenue streams,
- product roadmaps.

**What should a student be able to create at the end of the course?**

By the end of the course, the students should be able to create:

- a believable, detailed design future,
- a tech product prototype,

- a business model for the tech product,
- a minimum viable product (MVP),
- a tech product roadmap
- a pitch presentation for investors.

#### L.45.2.4 Course evaluation

**Table L.10:** Course grade breakdown

| Type | Default Points | Proposed Points |
|---|---|---|
| In-class work | 30 | 30 |
| Pitch presentation | 30 | 10 |
| Report | 0 | 10 |
| Exam | 40 | 50 |

**Late Submission Policy**

This policy will be strictly applied in this course. If a personal emergency should arise that affects your ability to turn in an assignment in a timely fashion, you must contact the course instructor BEFORE the deadline to get a "Special Late Submission Approval" from the course instructor. Without the "Special Late Submission Approval" submissions will be still accepted up to 48 hours late, but with a 50% penalty. No "Special Late Submission Approval" will be granted after the deadline. All late submissions should be submitted by email directly to the instructors.

**Cooperation Policy and Quotations**

We encourage vigorous discussion and cooperation in this class. You should feel free to discuss any aspects of the class with any classmates. However, we insist that any written material that is not specifically designated as a Team Deliverable be done by you alone. This includes answers to reading questions, individual reports associated with assignments, and labs. We also insist that if you include verbatim text from any source, you clearly indicate it using standard conventions of quotation or indentation and a note to indicate the source.

#### L.45.2.5 Grades range

**Table L.11:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

#### L.45.2.6 Resources and reference material:

Textbook(s)

- Tidd, J. & Bessant, J. (2011). Managing Innovation: Integrating Technological, Market and Organizational Change
- Bleecker, J. (2009). Design Fiction: A Short Essay om Design, Science, Fact and Fiction
- Stickdorn, M. & Schneider, J. (2010). This is Service Design Thinking. Wiley
- Brown, T. & Katz, B. (2009). Change by design. New York: Harper Business
- Osterwalder, A.& Pigneur, Y. (2010). Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers
- Sutherland, J. (2014). Scrum: The Art of Doing Twice the Work in Half the Time
- Ries, E. (2011). The Lean Startup.

Reference Materials

- Moore Geoffrey A. (1991) Crossing the Chasm - Harper Collins
- Moore Geoffrey A. (1995) Inside the Tornado - Harper Collins
- Ries Eric (2011). The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Publishing.

Computer Resources

- Students should have laptops with Internet access

## L.45.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table L.12:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Designing a Future: Defining Tech Trends | 10 |
| 2 | Designing a Tech Product Prototype | 12 |
| 3 | Designing a Business Model | 14 |
| 4 | Developing a Tech Product | 7 |
| 5 | Managing a Tech Product | 10 |
| 6 | Pitching / Presenting | 7 |

### L.45.3.1 Section 1

**Section title:** Designing a Future: Defining Tech Trends

**Topics covered in this section:**

- Emerging technologies/Gartner Hype Cycle
- Design Methods
- Design Fiction

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | no |
| Reports | yes |
| Essays | yes |
| Oral polls | yes |
| Discussions | yes |

**Typical questions for project presentation within this section**

1. What is the vision for the industry of choice?
2. How customers do their jobs in the industry right now?
3. What technologies are used by customers in the industry right now?
4. What technologies should be developed to fill in the existing tech niche?

**Typical questions for seminar classes (labs) within this section**

1. What is the Gartner Hype Cycle? How is it used by IT companies in to foresee the future?
2. What are the most important emerging technologies?
3. Why design methods are so popular today? What is their value?
4. What is the design fiction approach?
5. What is a diegetic prototype? What role does it play in design fiction contexts?

**Test questions for final assessment in this section**

1. Explain, the rationale behind the Gartner Hype Cycle.
2. How are diegetic prototypes used by companies to evaluate the technology?

### L.45.3.2 Section 2

**Section title:** Designing a Tech Product Prototype

**Topics covered in this section:**

- Design-thinking
- Empathy building
- Point-of-View
- Ideation
- Prototyping
- Testing

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Homework and group projects | yes |
| In-class assignment | yes |
| Presentations | yes |
| Discussions | yes |

**Typical questions for ongoing performance evaluation within this section**

1. Please, describe your persona. What does the typical day of your character look like?

2. Please, prepare the empathy map for your user.
3. Please, formulate the point of view for your user.
4. Please, conduct the ideation session, generate pool of ideas for future tech products and evaluate your ideas.
5. Please, design a tech product prototype and test it with your users.
6. Please, integrate the feedback into a tech product prototype.

**Typical questions for seminar classes (labs) within this section**

1. What is the difference between design and engineering approaches?
2. What are the properties of design-thinking?
3. What are the stages of the design-thinking methodology?
4. How can we develop the empathy with users?
5. What is a persona? How to design a persona?
6. How can we outline the most crucial pain points of the user?
7. How can we conduct the ideation sessions?
8. What prototyping methods are used by designers?
9. How can we test prototypes and collect the user feedback?

**Test questions for final assessment in this section**

1. What are the design methods used for empathy building
2. What are the ideation methods used for creating new tech products?
3. What are the prototyping methods used for creating the first representation of the tech product?

### L.45.3.3 Section 3

**Section title:** Designing a Business Model

**Topics covered in this section:**

- Innovation in Business Models
- Business Model Canvas
- Hypothesis and Experiments
- Minimum Viable Product (MVP)

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Homework and group projects | yes |
| In-class assignment | yes |
| Presentation | yes |
| Discussions | yes |

**Typical questions for project presentation within this section**

1. Please, develop the business model for your tech product.
2. Please outline all the hypotheses behind the business model and develop a plan how to test them through experiments.
3. Please, create the concept for your Minimum Viable Product.

**Typical questions for seminar classes (labs) within this section**

1. What business models are applied throughout the music industry history?
2. What is the value of the business model canvas by Alexander Osterwalder?
3. What are the components of the business model?
4. What are the business model assumptions? How can we test them?
5. What is the Minimum Viable Product (MVP)? How to define must-have, should-have and could-have requirements?

**Test questions for final assessment in this section**

1. What are the components of the business model?
2. What experiments can be conducted to test business models?
3. What is the approach to develop Minimum viable Product (MVP)?

### L.45.3.4 Section 4

**Section title:** Developing a Tech Product

**Topics covered in this section:**

- Agile Methods
- SCRUM Framework

- SCRUM Roles
- SCRUM Artifacts
- SCRUM Ceremonies

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Home assignment | no |
| In-class assignment | yes |
| Presentations | no |
| Discussions | yes |

**Typical questions for ongoing performance evaluation within this section**

1.
2.
3.
4.

**Typical questions for seminar classes (labs) within this section**

1. What is the difference between the waterfall planning and agile planning?
2. What are the SCRUM roles?
3. What are the SCRUM ceremonies?
4. What are the SCRUM artifacts?
5. How to scale SCRUM in the company?

**Test questions for final assessment in this section**

1. Please, discuss what are the responsibilities of SCRUM Master, SCRUM Team, Product Owner.
2. What is Sprint? How to do sprint planning?
3. What is Sprint Retrospective? How to conduct it?
4. What is Product Backlog and Sprint Backlog? How are they defined?

#### L.45.3.5 Section 5

**Section title:** Managing a Tech Product

**Topics covered in this section:**

- Product Manager Role
- Tech Product Critical Path
- Problem Space and Solution Space
- Business Motivation Model
- Stakeholder Management
- Product Roadmap
- Release Plan

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Homework and group projects | yes |
| In-class assignment | yes |
| Presentation | yes |
| Discussions | yes |

**Typical questions for project presentation within this section**

1. Please, develop the business motivation model for your company.
2. Please, develop the stakeholder communication plan.
3. Please, develop a product roadmap.

**Typical questions for seminar classes (labs) within this section**

1. Who is the product manager? What skills should the product manager have?
2. What is the tech product critical path?
3. What is the business motivation model? How is it used by the product manager?
4. How can we manage stakeholders of the product?
5. What is the difference between Problem Space and Solution Space?
6. What is the product roadmap? What is the release plan?

**Test questions for final assessment in this section**

1. What are the responsibilities of the product manager?
2. What are the types of stakeholders? How can we communicate with them?
3. How can we develop the product roadmap? What are the components?
4. What should be included in the release plan?

### L.45.3.6 Section 6

**Section title:** Pitching/ Presentation

**Topics covered in this section:**

- Pitch presentation
- Pitch components

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| In-class assignment | yes |
| Home assignment | no |
| Presentation | yes |
| Discussions | yes |

**Typical questions for project presentation within this section**

1. Please, prepare the pitch for potential investors.

**Typical questions for seminar classes (labs) within this section**

1. What is a pitch?
2. What makes a great pitch?
3. What sections should be included in the pitch presentation?

## L.46 Personal Efficiency Skills

- **Course name:** Personal Efficiency Skills
- **Course number:** N/A

- **Subject area:** Self-management

## L.46.1 Course Characteristics

## L.46.2 What subject area does your course (discipline) belong to?

### L.46.2.1 Key concepts of the class

- Personal Efficiency
- Time-management
- Proactivity
- Leadership
- Purposes and Values
- Negotiations
- Emotional Intellect
- Work in Teams
- Conflicts Solving

### L.46.2.2 What is the purpose of this course?

The main purpose is to teach students to be more efficient through:

- using time-management approaches
- proactive vision
- the ability to use different leadership styles
- projecting their purposes and understanding their values
- the ability to use “win-win” negotiations
- increasing their emotional intellect
- understanding their roles in team work
- the ability to solve different conflicts

### L.46.2.3 Course objectives based on Bloom’s taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize

- 7 habits of efficient people
- the concept of leadership
- the concept of emotional intellect
- the concept of conflict
- the concept of team
- the concept of negotiations

**- What should a student be able to understand at the end of the course?**

- the principles of time-management
- advantages and disadvantages of leadership styles
- the difference between reactivity and proactivity
- the difference between purposes and values
- good and bad effects of conflicts
- the difference between good and bad team

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- lead the list of tasks, projects and plan their day, week and month
- use different types of leadership due to the situation
- project their purposes
- use "win-win" negotiations
- understand and manage their emotions and emotions of other people
- work in teams in the most efficient way
- conflicts resolution

#### L.46.2.4 Course evaluation

#### L.46.2.5 Grades range

#### L.46.2.6 Resources and reference material

Textbook(s)

- Mihaly Csikszentmihalyi. Flow: The Psychology of Optimal Experience

**Table L.13:** Course grade breakdown

| Type | Default Points | Proposed Points |
|---|---|---|
| Classes attendance | | 45 |
| Work at classes | | 30 |
| Final Project | | 50 |

**Table L.14:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 90-125 |
| B. Good | 75-89 | 70-89 |
| C. Satisfactory | 60-74 | 50-69 |
| D. Poor | 0-59 | 0-49 |

- Stephen R. Covey. The 7 Habits of Highly Effective People
- Eliyahu M. Goldratt, Jeff Cox. "The Goal: A Process of Ongoing Improvement"
- Jen Sincero. You Are a Badass: How to Stop Doubting Your Greatness and Start Living an Awesome Life.
- Daniel Goleman. “Emotional Intelligence: Why It Can Matter More Than IQ”
- Ray Dalio. Principles: Life and Work
- Maksim Dorofeev. Jedi techniques
- Dave Logan, Halee Fischer-Wright, John King. Tribal Leadership: Leveraging Natural Groups to Build a Thriving Organization.
- Patrick Lencioni. The Five Dysfunctions of a Team: A Leadership Fable.
- Brett Blumenthal52 Small Changes for the Mind. Less Stress Increased Productivity Improved Memory and a Happier Mind

Computer Resources: none

## L.46.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

### L.46.3.1 Section 1

**Section title:** Be proactive

**Table L.15:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Be proactive | 1 |
| 2 | Begin with an end in mind | 3 |
| 3 | Put first things first | 8 |
| 4 | Think win-win | 8 |
| 5 | Seek first to understand, then to be understood | 8 |
| 6 | Synergize | 3 |
| 7 | Sharpen the saw. Don't work yourself to death | 1 |

### Topics covered in this section:

- proactivity reactivity
- reactivity

### What forms of evaluation were used to test students' performance in this section?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | no |
| Midterm evaluation | no |
| Testing (written or computer based) | no |
| Reports | no |
| Essays | yes |
| Oral polls | yes |
| Discussions | yes |

### Typical questions for ongoing performance evaluation within this section

1. Essay: "What makes me anxious?"

### Typical questions for seminar classes (labs) within this section

1. What is the difference between reactive and proactive people?
2. What is the circle of influence? What is the circle of concers? What is the difference between them?

### Test questions for final assessment in this section

1. What should I do to become proactive?

#### L.46.3.2 Section 2

**Section title:** Begin with an end in mind

**Topics covered in this section:**

- levels of priority
- purposes and values
- sense of life
- principles of life

**What forms of evaluation are used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | |
| no Homework and group projects | yes |
| Midterm evaluation | |
| Testing (written or computer based) | no |
| Reports | no |
| Essays | yes |
| Oral polls | yes |
| Discussions | yes |

**Typical questions for ongoing performance evaluation within this section**

1. Game in pairs "Why?"
2. Game in groups "The analysis of purposes"
3. Essay "My ideal day"

**Typical questions for seminar classes (labs) within this section**

1. What does happiness mean?
2. What are the stages of projecting purposes?
3. What does S.M.A.R.T. mean?
4. What are the principles of life mean? What principles do you have?

**Test questions for final assessment in this section**

1. What is the difference between purposes and values?
2. How to find out my values?

#### L.46.3.3 Section 3

**Section title:** Put first things first

**Topics covered in this section:**

- time-management

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | no |
| Reports | no |
| Essays | no |
| Oral polls | yes |
| Discussions | yes |

**Typical questions for ongoing performance evaluation within this section**

1. Project "How much time do I spent on interference?"
2. Homework "The analysis of 1 week of my time"

**Typical questions for seminar classes (labs) within this section**

1. What typical mistakes in the process of managing time do you know?
2. What types of tasks do you know?

**Test questions for final assessment in this section**

1. How to manage interference?
2. What ways of time-management do you know?

#### L.46.3.4 Section 4

**Section title:** Think win-win

**Topics covered in this section:**

- emotional bank account
- negotiations: integrative and hard
- leadership

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | no |
| Reports | no |
| Essays | no |
| Oral polls | yes |
| Discussions | yes |

**Typical questions for ongoing performance evaluation within this section**

1. Group game "Sticks"
2. The analysis of negotiations at video.
3. Game in pairs "Hard negotiations".
4. Game in pairs "Integrative negotiations".
5. The analysis of styles of leadership at video.
6. Games in pairs "Using different styles of leadership" .

**Typical questions for seminar classes (labs) within this section**

1. How to manage emotional bank account?
2. What styles of leadership do you know?

**Test questions for final assessment in this section**

1. What ways of hard negotiations do you know?
2. How to participate in integrative negoatiations?
3. What styles are the most suitable in different situations?

### L.46.3.5 Section 5

**Section title:** Seek first to understand, then to be understood

**Topics covered in this section:**

- emotional intellect

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | yes |
| Reports | no |
| Essays | no |
| Oral polls | yes |
| Discussions | yes |

**Typical questions for ongoing performance evaluation within this section**

1. Essay: "Which situations are most emotional for me?"
2. Test:"The level of my emotional intellect"

**Typical questions for seminar classes (labs) within this section**

1. Where do emotions come from?
2. What does emotional intellect mean?
3. What kinds of emotions do you know?

**Test questions for final assessment in this section**

1. What is the difference between emotional intellect and IQ?
2. Please, describe the model of emotional intellect?
3. What ways of managing emotions do you know?
4. What ways of understanding emotions do you know?

#### L.46.3.6 Section 6

**Section title:** Synergize

**Topics covered in this section:**

- work in teams
- conflicts resolution

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | yes |
| Midterm evaluation | no |
| Testing (written or computer based) | yes |
| Reports | no |
| Essays | no |
| Oral polls | yes |
| Discussions | yes |

**Typical questions for ongoing performance evaluation within this section**

1. Homework: please, assess the effectiveness of your team.
2. Test: which roles in team can you perform?
3. Game in groups: arguments for and against conflicts.

**Typical questions for seminar classes (labs) within this section**

1. What does "team" mean?
2. What are the characteristics of teams?
3. Stages of team's growth.
4. What roles in teams do you know?
5. What does "conflict" mean?
6. What is the formula of conflict?
7. What are the reasons of conflicts?
8. What types of conflicts do you know? What is the difference between them?

**Test questions for final assessment in this section**

1. What rules to build a great team do you know?
2. Which roles in teams are obligatory and which are not?
3. What strategies of behavior in conflicts do you know? How to use them? Which strategy is the most preferable in which situations?

### L.46.3.7 Section 7

**Section title:** Sharpen the saw. Don't work yourself to death

**Topics covered in this section:**

- spheres of recreation

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | no |
| Homework and group projects | no |
| Midterm evaluation | no |
| Testing (written or computer based) | no |
| Reports | no |
| Essays | no |
| Oral polls | yes |
| Discussions | yes |

**Typical questions for ongoing performance evaluation within this section**

1. Essay: how can I sharpen my saw?

**Typical questions for seminar classes (labs) within this section**

1. In which spheres of recreation can you participate?

**Test questions for final assessment in this section**

1. How can you sharpen your saw?

# Appendix M

# Catalogue of the core courses

The core courses administered in the programs are:

1. Advanced Information Retrieval (see Section M.1 on page 616)
2. Advanced Machine Learning (see Section **??** on page **??**)
3. Advanced Networking (see Section **??** on page **??**)
4. Advanced Robotics (see Section M.4 on page 639)
5. Advanced Security (see Section M.5 on page 647)
6. Advanced Statistics (see Section M.6 on page 654)
7. Analysis of Software Artifacts (see Section M.7 on page 662)
8. Architectures for Software Systems (see Section M.8 on page 676)
9. Behavioral and Cognitive Robotics (see Section **??** on page **??**)
10. Big Data Technologies and Analytics (see Section M.10 on page 692)
11. Classical Internet Applications (see Section M.11 on page 700)
12. Communication (see Section M.12 on page 707)
13. Computational Intelligence (see Section M.13 on page 711)
14. Computer Vision (see Section **??** on page **??**)
15. Cybercrime and Forensics (see Section M.15 on page 725)
16. Distributed Systems (see Section **??** on page **??**)
17. Dynamics of Nonlinear Robotic Systems (see Section **??** on page **??**)
18. High-Dimensional Data Analysis (see Section M.18 on page 750)
19. Large Installation Administration (see Section **??** on page **??**)
20. Machine Learning (see Section **??** on page **??**)

21. Managing Software Development (see Section M.21 on page 769)
22. Metrics and Empirical Methods for Software Engineers and Data Scientists (see Section **??** on page **??**)
23. Models of Software Systems (see Section M.23 on page 787)
24. Neuroscience (see Section M.24 on page 795)
25. Offensive Technologies (see Section M.25 on page 807)
26. Optimization (see Section M.26 on page 816)
27. Personal Software Process (see Section M.27 on page 819)
28. Requirements Engineering (see Section M.28 on page 827)
29. Security of System and Networks (see Section M.29 on page 837)
30. Sensing, Perception, and Actuation (see Section M.30 on page 845)

They are divided across programs as follows:

- **Software Engineering**
  1. Analysis of Software Artifacts (see Section M.7 on page 662)
  2. Architectures for Software Systems (see Section M.8 on page 676)
  3. Communication (see Section M.12 on page 707)
  4. Managing Software Development (see Section M.21 on page 769)
  5. Requirements Engineering (see Section M.28 on page 827)
  6. Metrics and Empirical Methods for Software Engineers and Data Scientists (see Section **??** on page **??**)
  7. Models of Software Systems (see Section M.23 on page 787)
  8. Personal Software Process (see Section M.27 on page 819)
- **Robotics**
  1. Advanced Robotics (see Section M.4 on page 639)
  2. Behavioral and Cognitive Robotics (see Section **??** on page **??**)
  3. Computational Intelligence (see Section M.13 on page 711)
  4. Computer Vision (see Section **??** on page **??**)
  5. Dynamics of Nonlinear Robotic Systems (see Section **??** on page **??**)
  6. Machine Learning (see Section **??** on page **??**)
  7. Neuroscience (see Section M.24 on page 795)
  8. Sensing, Perception, and Actuation (see Section M.30 on page 845)

- **Data Science**
  1. Advanced Information Retrieval (see Section M.1 on page 616)
  2. Advanced Machine Learning (see Section **??** on page **??**)
  3. Advanced Statistics for DS (see Section M.6 on page 654)
  4. Big Data Technologies and Analytics (see Section M.10 on page 692)
  5. High-Dimensional Data Analysis (see Section M.18 on page 750)
  6. Machine Learning (see Section **??** on page **??**)
  7. Metrics and Empirical Methods for Software Engineers and Data Scientists (see Section **??** on page **??**)
  8. Optimization (see Section M.26 on page 816)
- **Secure Systems and Network Engineering**
  1. Advanced Networking (see Section **??** on page **??**)
  2. Classical Internet Applications (see Section M.11 on page 700)
  3. Distributed Systems (see Section **??** on page **??**)
  4. Security of System and Networks (see Section M.29 on page 837)
  5. Cybercrime and Forensics (see Section M.15 on page 725)
  6. Advanced Security (see Section M.5 on page 647)
  7. Large Installation Administration (see Section **??** on page **??**)
  8. Offensive Technologies (see Section M.25 on page 807)

## M.1 Advanced Information Retrieval

- **Course name:** Advanced Information Retrieval
- **Course number:** N/A

### M.1.1 Course Characteristics

### M.1.2 What subject area does your course (discipline) belong to?

Computer systems organization; Information systems; Real-time systems; Information retrieval; World Wide Web

#### M.1.2.1 Key concepts of the class

- Data indexing
- Relevance and ranking

#### M.1.2.2 What is the purpose of this course?

The course is designed to prepare students to understand and learn contemporary tools of information retrieval systems. Students, who will later dedicate their engineering or scientific careers to implementation of search engines, social networks, recommender systems and other content services will obtain necessary knowledge and skills in designing and implementing essential parts of such systems.

#### M.1.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize

- Terms and definitions used in area of information retrieval,
- Search engine and recommender system essential parts,
- Quality metrics of information retrieval systems,
- Contemporary approaches to semantic data analysis,
- Indexing strategies.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- How to design a recommender system from scratch,
- How to evaluate quality of a particular information retrieval system,
- Core ideas and system implementation and maintenance,
- How to identify and fix information retrieval system problems.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Build a recommender service from scratch,
- Implement proper index for an unstructured dataset,
- Plan quality measures for a new recommender service,
- Run initial data analysis and problem evaluation for a business task, related to information retrieval.

#### M.1.2.4 Course evaluation

**Table M.1:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 0 |
| Interim performance assessment | 30 | 0 |
| Assessments | 0 | 50 |
| Exams | 50 | 50 |

If necessary, please indicate freely your course's features in terms of students' performance assessment:

Labs are followed by the home works. Home works are covering 50% of the grade. 50% of the grade fall to exam session, which will be in the form of project defence.

#### M.1.2.5 Grades range

#### M.1.2.6 Resources and reference material

Main textbook:

**Table M.2:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 60-79 |
| C. Satisfactory | 60-74 | 40-59 |
| D. Poor | 0-59 | 0-39 |

- "An Introduction to Information Retrieval" by Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Cambridge University Press (any edition)

Other reference material:

- “Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit,” Steven Bird, Ewan Klein, and Edward Loper. `https://www.nltk.org/book/`[link]

### M.1.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.3:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Text processing (1, 3, 4, 7, 8, 9) | 24 |
| 2 | Text indexing and search (2, 5) | 8 |
| 3 | Web basics and algorithms for web (6, 10, 14) | 12 |
| 4 | Media types processing: images, video, sound (11, 12) | 8 |
| 5 | Quality assessment (13, 15) | 8 |

#### M.1.3.1 Section 1

**Section title:** Text processing

**Topics covered in this section:**

- Introduction to information retrieval
- Distributional semantics. Vector model. Dimension reduction

- ML approaches to vector modelling
- Spellchecking and query correction
- Query expansion and suggest
- Language model. Topic model. Clustering and classification

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Build term-document matrix from the text.
2. Visualize a dataset in 2D using PCA.
3. Find K most similar text using ANN-based language model (embeddings).
4. Implement Levenstein distance function.
5. Discover 5 major topics in a text dataset.

**Typical questions for seminar classes (labs) within this section**

1. Build term-document matrix of a set of web pages using TF-IDF.
2. Implement PCA-based latent semantic analysis for a set of text with labels. Visualize classes (labels) in 2D.
3. Train Doc2vec model and implement search with ranking using cosine similarity.
4. Implement spellchecking algorithm.
5. Discover optimal number of topics in a dataset using topic modelling.

**Test questions for final assessment in this section**

1. What are the approaches to vector space modelling for text datasets?
2. What are the most common techniques to process text before indexing?
3. Explain topic modelling problem statement.

#### M.1.3.2 Section 2

**Section title:** Text indexing and search

**Topics covered in this section:**

- Building inverted index.
- Language, tokenization, stemming, searching, scoring.
- Indexing for vector model.
- Kd-trees, Quad-trees, Annoy, FAISS, HNSW index.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Build inverted index for a text.
2. Tokenize a text.
3. Write kd-tree.
4. Implement small-world graph.
5. Build HNSW index.

**Typical questions for seminar classes (labs) within this section**

1. Build inverted index for a set of web pages.
2. build a distribuition of stems/lexemes for a text.
3. Choose and implement persistent index for a given text collection.
4. Compare 2 logarithmic index structures for a given task.
5. Choose and implement updateable index for vector space of a given dimensions.

**Test questions for final assessment in this section**

1. Explain how (and why) KD-trees work.

2. What are weak places of inverted index?
3. State the problem of approximate nearest neighbours search. Overview solutions.
4. Discuss HNSW index architecture.

#### M.1.3.3 Section 3

**Section title:** Web basics and algorithms for web

**Topics covered in this section:**

- Web basics. Internet crawling. XML and HTML processing. Dynamic documents processing.
- On newsfeeds.
- Web search specific topics. PageRank. Duplicates. CTR.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Extract text from web page.
2. Extract text from dynamic web page.
3. Crawl a website with respect of robots.txt.
4. Explain PageRank.

**Typical questions for seminar classes (labs) within this section**

1. Write a crawler for a dynamically generated web site.
2. Build a graph of a website.
3. Run a PageRank on a graph of a single website.
4. Compute CTR for a given log and train a relevance model using this data.

**Test questions for final assessment in this section**

1. Explain what is DOM (document object model) and how is it used in document parsing?
2. What is the difference in crawling static and dynamic pages?
3. What are the weaknesses of PageRank?

### M.1.3.4 Section 4

**Section title:** Media types processing: images, video, sound

**Topics covered in this section:**

- Image and video processing.
- Image understanding
- Image enhancing
- Video understanding
- Audio processing
- Speech-to-text

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Extract 4 major colors from image.
2. Split video into shots.
3. Identify a chord in audio.
4. Text-2-speech.
5. Label objects in images.

**Typical questions for seminar classes (labs) within this section**

1. Build a "search by color" feature.
2. Extract scenes from video.
3. Build a voice fingerprint and identify a person by voice.
4. Write a voice-controlled search.
5. Semantic search withing unlabelled image dataset.

**Test questions for final assessment in this section**

1. What the the approached to image understanding?
2. How to cluster a video into scenes and shots?
3. How speech-to-text technology works?
4. How to build audio fingerprints?

### M.1.3.5 Section 5

**Section title:** Quality assessment

**Topics covered in this section:**

- Search quality assessment.
- Recommender quality assessment.
- A/B testing.
- SBS.
- pFound, DCG, nDCG.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Propose a strategy for A/B testing.
2. Propose recommender quality metric.
3. Implement DCG metric.
4. Discuss relevance metric.

**Typical questions for seminar classes (labs) within this section**

1. Provide a framework to accept/reject A/B testing results.
2. Compute DCG for an example query for random search engine.
3. Implement a metric for a recommender system.
4. Implement pFound.

**Test questions for final assessment in this section**

1. What is SBS (side-by-side) and how is it used in search engines?
2. Compare pFound with CTR and with DCG.
3. Explain how A/B testing works.

## M.2 Advanced Machine Learning

- **Course name:** Advanced Machine Learning
- **Course number:** DS-02

### M.2.1 Course Characteristics

#### M.2.1.1 Key concepts of the class

- Advanced topics in machine learning
- To develop research interest in the theory and application of machine learning

#### M.2.1.2 What is the purpose of this course?

This course is designed for graduate students to provide comprehensive and advance topics in machine learning. Student will learn to implement the machine learning models in Python programming environment from data science prospective. In this course, we will cover Neural Network Architectures (CNN, LSTM, Attention, RNN), Regularization, Genetic algorithm, Graphical Models (HMM), Generative models, Reinforcement learning, Collaborative filtering and recent trends in machine learning. The end of the day they will able to apply machine learning algorithms to solve real-world problems.

#### M.2.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to apply machine learning models in real-world applications.

- Understanding of in depth details about the machine learning models
- Solve the problem in hand by apply machine learning models
- Ability to analyze the models capacity and performances.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to choose the correct machine learning model.

- Suitability of different machine learning models in different scenarios
- Ability to choose the right model for the given problem

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to deploy and developed models.

- Hands on experience to implement different models to know inside behavior
- Sufficient exposure to train and deploy model for the given task
- Fine tune the deployed model in the real-world settings

#### M.2.1.4 Course evaluation

**Table M.4:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 20 |
| Interim performance assessment | 30 | 50 |
| Exams | 50 | 30 |

#### M.2.1.5 Grades range

**Table M.5:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 90-100 |
| B. Good | 75-89 | 75-89 |
| C. Satisfactory | 60-74 | 60-74 |
| D. Poor | 0-59 | 0-59 |

#### M.2.1.6 Resources and reference material

- Handouts supplied by the instructor
- Materials from the interment and research papers shared by instructor
-
-
-

### M.2.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.6:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Neural Networks | 16 |
| 2 | Graphical Model and Evolutionary Computation | 12 |
| 3 | Collaborative Filtering and Generative Models | 8 |
| 4 | Reinforcement Learning | 8 |

#### M.2.2.1 Section 1

**Section title:** Neural Networks

**Topics covered in this section:**

- Neural Network Architectures
- Recurrent Neural Networks
- LSTM and Attention

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. What is the role of activation function in artificial neural network?
2. How Recurrent neural network can efficiently utilize the sequential information?
3. How you will evaluate the models that it learns the concept correctly?
4. How you distinguish different neural networks?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for seminar classes (labs) within this section**

1. Implementation of neural networks for classification and regression problems
2. Implement neural networks of natural language processing task
3. Fine tune the neural networks for given problem
4. Find the optimal hyper-parameters to converge the neural network

**Test questions for final assessment in this section**

1. Calculate the parameters for the given dataset
2. Analyze the problem and choose the correct neural network for the given problem
3. Explanation of the model selection and designed architecture
4. Different learning rates and initialization
5. Batch processing and GPU consideration

#### M.2.2.2 Section 2

**Section title:** Graphical Models and Evolutionary Computation

**Topics covered in this section:**

- Hidden Markov Model
- Decoding and inferencing in graphical models
- Evolutionary Computation - Genetic Algorithm

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Calculate the probabilities for making decisions
2. Draw the graphical models for given problem
3. Perform the classification task through graphical models

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

4. Comparison of the models and gold standard test

**Typical questions for seminar classes (labs) within this section**

1. Implement graphical models
2. Perform decoding task
3. Implement inferencing task for the given problem
4. Calculate the performance measures
5. Analyze the model performance.

**Test questions for final assessment in this section**

1. Design and analyze the graphical models
2. Dry run the model to make decisions for classification tasks
3. Calculate the performance measures
4. Perform the analysis over the given dataset

### M.2.2.3 Section 3

**Section title:** Collaborative Filtering and Generative Models

**Topics covered in this section:**

- Collaborative Filtering
- Simple GAN
- Conditional GAN
- Time-series prediction

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. how to find the equilibrium point in GANs?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

2. How you can define discriminator and generator?
3. How you can make suitable setting to train the model correctly?
4. How you will evaluate your model?
5. What are the effects of different loss functions?

**Typical questions for seminar classes (labs) within this section**

1. Implement GAN for given task
2. Implement conditional GAN
3. Analysis of different loss functions
4. Experiments to look-inside for network convergence

**Test questions for final assessment in this section**

1. Design and define the generative models for the given task
2. What are the possible solutions to make the training fast?
3. How you can initialize the data distribution and its impact.
4. What are the possible solutions if discriminator is two strong?

#### M.2.2.4 Section 4

**Section title:** Reinforcement Learning

**Topics covered in this section:**

- Value-based Reinforcement Learning
- Policy-based Reinforcement Learning
- Q-Learning
- Policy Gradients
- Deep Q Network (DQN)

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are the different kinds of Reinforcement Learning?
2. What is the relationship between the agent and state?
3. How you define fully and partially observed states?
4. Explain Markov decision process in fully observed situations

**Typical questions for seminar classes (labs) within this section**

1. Implement Q learning algorithm
2. Analyze the performance of DQN
3. Understand the loss functions and network architecture details
4. Apply reinforcement learning to a given task

**Test questions for final assessment in this section**

1. Calculate the next move of the agent in fully observed state
2. Explain the model outcome in reinforcement learning scenarios
3. What kind of tasks we can solve through reinforcement learning models?
4. Calculate the value of an agent for the next move

# M.3 Advanced Networking

- **Course name:** Advanced Networking
- **Course number:** xyz

## M.3.1 Course Characteristics

### M.3.1.1 Key concepts of the class

- Principles of designing and implementing advanced networking concepts
- Developing research interest in the modern network applications

### M.3.1.2 What is the purpose of this course?

Advanced Networking is the core course for SNE program that provides the students with advanced knowledge about computer networks. The course starts with advanced networking topics such as quality of service (QoS), Multiprotocol Label Switching, and software defined networking (SDN). Furthermore, this course also covers some state of the art networking paradigms such as 5G Wireless Networks, and content delivery network (CDN). This course includes hands-on exercise and the students will practice their skills on the real hardware and software. Students will have the opportunity to apply the knowledge obtained in the class, on real hardware.

### M.3.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to understand advanced networking concepts

- Key principles involved in designing and implementing Quality of Service (QoS) in existing networks
- Multi protocol Label Switching (MPLS) and its variants
- Virtual private networks and its implementation
- Wireless networks
- Optical networks

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to understand the key approaches and techniques for enabling advanced networking services

- Circuit switched and packet switched networks
- Virtual Private Networks (VPNs) and its variants
- Connectivity for mobile users
- State of Art and future wireless networks

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to develop and implement the followings:

- Implementation of QoS in existing networks
- Designing and implementing software defined networks based routing policies
- Orchestration and management of networks
- Designing resource sharing approaches for scarce wireless resources in wireless networks

#### M.3.1.4 Course evaluation

**Table M.7:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 20 |
| Interim performance assessment | 30 | 50 |
| Exams | 50 | 30 |

#### M.3.1.5 Grades range

#### M.3.1.6 Resources and reference material

-
-

**Table M.8:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 90-100 |
| B. Good | 75-89 | 75-89 |
| C. Satisfactory | 60-74 | 60-74 |
| D. Poor | 0-59 | 0-59 |

-
-
- Recent research papers and online materials

## M.3.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.9:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction of advanced networking | 20 |
| 2 | Advanced networking concepts | 20 |
| 3 | Wireless and future Networks | 16 |

### M.3.2.1 Section 1

**Section title:** Introduction of advanced networking concepts

**Topics covered in this section:**

- Quality of Service in networks
- Multi protocol Label Switching (MPLS)
- Optical Networks
- Content and Multimedia

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. What is the major issue in providing Quality of Service (QoS) in existing networks?
2. Why QoS has become essential in current networks recently?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

3. State the types of delays a packet experiences in a network?
4. Define transmission delays and the approaches to control it?
5. Briefly explain about the overprovisioning technique to enhance the QoS in networks?
6. What is the effect of attenuation in optical communication and how to resolve this problem?

**Typical questions for seminar classes (labs) within this section**

1. Test and deploy your QoS rules to prioritize the downloading of a file (or any other scenario) over the bandwidth test.
2. Try to set QoS rules to traffic allocation once and then set it to priority-based QoS, what are the differences?
3. What are CoS, ToS, Diffserv, DSCP, DS?
4. What is LDP, LSP? and how can you list them?
5. Does VPLS require disabling PHP?

**Test questions for final assessment in this section**

1. State the problem that can arise if we use global labels for MPLS?
2. How does shaping of traffic enhance the QoS in networks?
3. What is policing and its benefits
4. What is jitter and what types of applications are the most affected by jitter? Propose a solution to overcome the jitter problem?
5. Briefly explain Integrated services (IntServ) and differentiated services (Diffserv)?
6. Briefly explain what is Optical burst switching and its benefits compared to optical circuit switching?

#### M.3.2.2 Section 2

**Section title:** Advanced networking concepts

**Topics covered in this section:**

- Software-defined networking (SDN)
- Network Function Virtualization (NFV)
- Virtualized Networks and network slicing
- Orchestration and management of Networks (MANO platform)
- Network Mobility and mobile IP

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is virtual network function?
2. Define the benefits achieved by virtualizing the networks?
3. State the benefits of reactive and proactive models for populating OpenFlow switch tables?
4. How SDN enables centralized control in the networks?

**Typical questions for seminar classes (labs) within this section**

1. What is NVF? and what is the main goal of using it?
2. What kind of problem does NVF solve?
3. What is the difference between SDN and NFV?
4. What are the pros and cons of NVF anf SDN?
5. What are the security challenges in SDN ?

**Test questions for final assessment in this section**

1. State benefits of using SDN controllers in exiting networks?
2. What is a network slice and state its benefits?
3. How does the MANO platform enable management and orchestration of networks?
4. State the pros and cons of virtualizing the networks?

#### M.3.2.3 Section 3

**Section title:** Wireless and future Networks

**Topics covered in this section:**

- Wireless Local area network
- Bluetooth Networks
- Cellular Networks and its generations
- 5G and beyond networks
- Information-centric networking (ICN)

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. State the challenges of wireless networks?
2. State the reasons of using CSMA/CA in wireless networks?
3. Why do we need a back off timer in IEEE 802.11 networks?
4. What is the difference between OFDMA and OFDM?

**Typical questions for seminar classes (labs) within this section**

1. Simulate an IEEE 802.11b network.

2. Simulation of resource sharing for different networking services in 5G.
3. Performance analysis of wireless networks in terms of streaming applications.

**Test questions for final assessment in this section**

1. Briefly explain about the Exposed node problem. Additionally, discuss how IEEE 802.11 addresses this problem?
2. State the usage of short Inter-frame spacing (SIFS) in IEEE 802.11?
3. Briefly explain the usage of RSVP-TE and what additional information is present in RSVP-TE compared to RSVP?
4. Briefly explain about the Hidden node problem. Additionally, discuss how IEEE 802.11 addresses this problem?

## M.4 Advanced Robotics

- **Course name:** Advanced Robotics
- **Course number:**

### M.4.1 Course characteristics

### M.4.2 What subject area does your course (discipline) belong to?

Robotic control.

### M.4.3 Key concepts of the class

- Elastostatic modeling and calibration of robots
- Advanced control approaches for compliant robotic systems

### M.4.4 What is the purpose of this course?

While traditional robotics studies rigid robots and manipulators, many practical robotic systems exhibit non-negligible compliance. Its effects can be both detrimental (for instance, decrease in positioning accuracy of industrial manipulators) and beneficial (improved safety during human-robot interaction), depending on the application. However, regardless of whether the robot's compliance is positive or negative, it must be accurately accounted for during modeling, trajectory tracking and robot control tasks. The main purpose of this course is to introduce elastostatic modeling of manipulators and robotic systems, methods for calibration of these devices, as well as advanced approaches to control robotic systems with non-negligible stiffness.

### M.4.5 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember

- How to derive expressions for position kinematics and differential kinematics of serial manipulators,

- What approaches exist to model robot joints' elasticity,
- How to model dynamics of compliant robots,
- Fundamental principles of position tracking control for robots with compliance,
- Motivation behind energy-based approaches to control elastic robots.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- How to find Jacobian for series and parallel robots and use it to compute forces and torques,
- What constitutes a common manipulator calibration procedure,
- Reasons and examples of singularities for serial and parallel robots,
- How to drive elastic robots into limit cycles and what benefits does it bring in terms of control effort,
- How to model and control tendon-driven robots.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Find stiffness matrix for given manipulator,
- Analyze joint constraints and find singularities,
- Perform robot calibration procedure,
- Apply passivity principle to design stable position controllers,
- Design force controller for elastic and compliant robots.

### M.4.6 Course evaluation

**Table M.10:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 10 |
| Interim performance assessment | 30 | 60 |
| Exams | 50 | 30 |

If necessary, please indicate freely your course's features in terms of students' performance assessment:

The course grades are given according to the following rules: In-class discussion and lab performance = 10 pts, Homework assignments (4) = 20 pts, Quizzes (4) = 20 pts, Exams = 30 pts, Term project = 20 pts.

### M.4.7 Grades range

**Table M.11:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

If necessary, please indicate freely your course's grading features.

### M.4.8 Resources and reference material

Textbooks:

- 
- 
- 
- 
- 
- 

### M.4.9 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

#### M.4.9.1 Section 1

**Section title:** Stiffness modeling

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Stiffness modeling | 6 |
| 2 | Robot calibration | 6 |
| 3 | Position tracking | 6 |
| 4 | Energy, impedance, and force control | 6 |

**Topics covered in this section:**

- Position and velocity kinematics
- Virtual joint modeling
- Finite element analysis
- Matrix structural analysis

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Name types of robot workspace.
2. Name key features and differences between serial and parallel manipulators.
3. What is Jacobian matrix and how to use it for singularity analysis.
4. What is stiffness matrix of manipulator and what does it describe?

**Typical questions for seminar classes (labs) within this section**

1. Find stiffness matrix of a given parallel robotic platform.
2. Apply direct FEA method to analyze compliance of a given manipulator.
3. Perform matrix structural analysis of a cantilever beam.
4. Find stiffness matrix of a two-link manipulator with elastic joint.
5. Model stiffness of a non-rigid mobile platform.

**Test questions for final assessment in this section**

1. Describe main stiffness modeling approaches, their particularities, advantages and limitations.

2. Use variable joint model for a serial manipulator (assume all elements are flexible) to find stiffness matrix.
3. Drive VSJ and MSA models of the tripteron robot shown.

### M.4.9.2 Section 2

**Section title:** Robot calibration

**Topics covered in this section:**

- Types of robot calibration
- Sources of uncertainties and model errors in practical robots
- Robot errors
- Complete, irreducible geometric models
- Elastostatic calibration

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Why is robot calibration needed?
2. What are the main sources of errors in robot parameters?
3. Give examples of geometric and non-geometric errors.
4. Describe typical steps of calibration procedure.

**Typical questions for seminar classes (labs) within this section**

1. Drive information matrix of a 2-link manipulator.
2. Estimate identification accuracy for 4-link manipulator.
3. Comment on differences between compliance matrix of a manipulator obtained via CAD modeling and identification results.
4. Perform model reduction for a given manipulator.

**Test questions for final assessment in this section**

1. Describe particularities and difficulties of the elastostatic calibration.
2. What do good/bad accuracy and repeatability mean?
3. What is complete, irreducible geometric model and why do we need it?
4. Find complete and irreducible model for geometric calibration of robot presented below.

### M.4.9.3 Section 3

**Section title:** Position tracking

**Topics covered in this section:**

- Adaptive control of flexible joint manipulators
- Adaptive robust control
- Modeling and control of cable-driven robotic systems

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What challenges does robot compliance pose for a control system?
2. What are the mathematical fundamentals of adaptive control?
3. How does cable elasticity affect dynamics of tendon-driven robots?
4. How to perform feedback linearization for a given compliant robot?

**Typical questions for seminar classes (labs) within this section**

1. Design PD controller with gravity compensation for a manipulator with elastic joint.
2. Numerically model behavior of compliant robot with nonlinear controller.

3. Numerically model and compare accuracy and power efficiency of robust and adaptive controllers for a cable-driven robot.
4. Analyze stability of adaptive controller.

**Test questions for final assessment in this section**

1. Provide examples of practical systems with non-collocated feedback. What unique challenges does this pose for control systems?
2. Design a position tracking controller for a given compliant system.
3. Analyze stability of a given nonlinear control approach.

### M.4.9.4 Section 4

**Section title:** Energy, impedance, and force control

**Topics covered in this section:**

- Energy-based control of compliant robots
- Limit cycles
- Passivity-based control
- Impedance control

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Provide examples of passive and active systems.
2. What are limit cycles?
3. What components of mechanical energy exist in robots with compliance?
4. What happens with the energy of passive systems with time?

**Typical questions for seminar classes (labs) within this section**

1. Find limit cycles of a given robot with compliance.
2. Design gravity and compliance compensator for a robot with flexible joints.
3. Simulate numerically behavior of a compliant robot during cyclic motion.
4. Implement and simulate passivity-based control over given robot.

**Test questions for final assessment in this section**

1. What are the physical fundamentals behind the concept of passivity and passivity-based control?
2. Drive the dynamics of a given elastically actuated robot.
3. Analyze stability of a given system with passivity-based controller.

## M.5 Advanced security

- **Course name:** Advanced Security
- **Course number:**

### M.5.1 Course characteristics

#### M.5.1.1 Key concepts of the class

- Wireless network security
- Software security

#### M.5.1.2 What is the purpose of this course?

In this course, server based and application based web attacks are thought in a simulated/test environment. We begin by exploring advanced techniques and attacks to which all modern-day complex applications may be vulnerable. We'll learn about new web frameworks and web backends, then explore encryption as it relates to web applications, digging deep into practical cryptography used by the web, including techniques to identify the type of encryption in use within the application and methods for exploiting or abusing it.

#### M.5.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define

- RF and Bluetooth security
- Database security/Wifi security
- Web vulnerabilities
- API security
- Software security
- Network security

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- Web vulnerabilities
- WPA and WEP definition and differences 8
- Use of OWASP
- Software security

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply

- Track application security against known standard OWASP and SANS categories
- To perform Bluetooth sniffer test.
- WPA implementation
- Testing security features within applications

### M.5.1.4 Course evaluation

**Table M.12:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 50 |
| Interim performance assessment | 30 | 0 |
| Exams | 50 | 50 |

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

### M.5.1.5 Grades range

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table **??**, but it may change slightly (usually reduced) depending on how the semester progresses.

**Table M.13:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

#### M.5.1.6 Resources and reference material

- Michael Sikorksi and Andrew Honig, *Practical Malware Analysis by, 1nd Edition, Kindle Edition*

### M.5.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.14:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Bluetooth security | 12 |
| 2 | Wifi security | 12 |
| 3 | Web security | 12 |
| 4 | Network security | 12 |

#### M.5.2.1 Section 1

**Section title:**

Bluetooth security

**Topics covered in this section:**

- Bluetooth standard
- Bluetooth applications
- Bluetooth security

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Explain an ad hoc networking

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

2. Explain frequency hoping
3. Establishing Piconets example
4. What is sniff mode?

**Typical questions for seminar classes (labs) within this section**

1. What are possible security flaws for common applications of the bluetooth technology?
2. What approaches are used to increase radio channel security and throughput?
3. How radio waves are propagated through environment?

**Test questions for final assessment in this section**

1. How radio waves are propagated through environment?
2. What is park mode?
3. How Service Discovery Protocol works?

#### M.5.2.2 Section 2

**Section title:**

Wifi security

**Topics covered in this section:**

- Eavesdropping
- DoS
- WEP / WPA / RSN

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Define WEP issues!

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

2. Describe active attack.
3. TKIP?
4. AES-CCMP?

**Typical questions for seminar classes (labs) within this section**

1. Handoff-iapp (802.11f)
2. Pre-auth (802.11i)
3. EduRoam

**Test questions for final assessment in this section**

1. EduRoam?
2. Explain TLS handshake.
3. Differences between WEP and WPA?

#### M.5.2.3 Section 3

**Section title:**

Web security

**Topics covered in this section:**

- Security related web technologies.
- Same Origin Policy.
- Web Attacker Model

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. What is Same Origin Policy?
2. To which attack does SOP mitigate?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

3. How the Document object model could be used for an attacker to manipulate the web browser data?

**Typical questions for seminar classes (labs) within this section**

1. Vulnerability analysis and exploitation for a given web application.
2. Write and deploy a WAF rules to mitigate a spicific web attack.

**Test questions for final assessment in this section**

1. Vulnerability analysis and exploitation for a given web application, explain.
2. How the Document object model could be used for an attacker to manipulate the web browser data?

#### M.5.2.4 Section 4

**Section title:**

Network security

**Topics covered in this section:**

- Network security policies and practices
- Nmap
- VPN
- IPsec

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. What is VPN?
2. IPsec?
3. Nmap?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

### Typical questions for seminar classes (labs) within this section

1. Exploit the difference between VPN and sock5.
2. Exploit IPsec effects on the performance of a networking device.
3. Why does nmap produce false-positive when scanning a windows host?
4. Can you improve nmap scanning technique? Elaborate!

### Test questions for final assessment in this section

1. What are the difference between VPN and sock5
2. Does IPsec effects the performance of a networking device?
3. What does nmap produce when scanning a windows host?
4. Can you improve the scanning technique?

## M.6 Advanced Statistics

- **Course name:** Advanced Statistics
- **Course number:** DS-03

### M.6.1 Course Characteristics

#### M.6.1.1 Key concepts of the class

- Statistical inference
- Non parametric statistics
- Test of statistical hypotheses
- Simple linear regression and correlation analysis

#### M.6.1.2 What is the purpose of this course?

The main purpose of this course is to present the fundamentals of inferential statistics to the future software engineers and data scientists, on one side providing the scientific fundamentals of the disciplines, and on the other anchoring the theoretical concepts on practices coming from the world of software development and engineering. The course covers the statistical analysis of data with limited assumptions on the distribution, with reference to testing hypotheses, measuring correlations, building samples, and performing regressions.

#### M.6.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to:

- Remember the fundamentals of inferential statistics
- Remember the specifics and purpose of different hypothesis tests
- Distinguish between parametric and non parametric tests

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to understand:

- the basic concepts of inferential statistics
- the fundamental laws in statistics
- the concept of null and alternative hypotheses
- the hypotheses test procedure

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

- To understand the problems related to analyse statistically data not distributed normally
- To know the more recent computationally-intensive techniques that can help to describe samples and to infer properties of populations in absence of normality
- To identify situations when the data is on nominal scales so alternative techniques should be use, and act accordingly.
- To be able to run experiment to evaluate hypotheses for situation of scarce data, distributed non normally, on different kinds of scales.

#### M.6.1.4 Course evaluation

**Table M.15:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Weekly quizzes | ? | 10 |
| Midterm | ? | 20 |
| Final oral exam | ? | 35 |
| Final written exam | ? | 30 |
| Participation | ? | 5 |

#### M.6.1.5 Grades range

**Table M.16:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 95-100 |
| B. Good | 75-89 | 75-94 |
| C. Satisfactory | 60-74 | 55-74 |
| D. Poor | 0-59 | 0-54 |

#### M.6.1.6 Resources and reference material

- Wasserman L. (2006) All of Nonparametric Statistics. Springer
- Randles, R.H. and Wolfe, D.A. (1991). Introduction to the Theory of Nonparametric Statistics. Melbourne: Robert Krieger. (Ch.1-Ch.4)
- Hastie, T. Tibshirani, R. and Friedman, J. (2008) The Elements of Statistical Learning 2ed. Springer
- Hollander, M. and Wolfe, D.A. (1999). Nonparametric Statistical Methods, 2nd ed. New York: John Wiley.

### M.6.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.17:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Sampling Distributions Associated with the Normal Population | 15 |
| 2 | Test of Statistical Hypotheses | 30 |
| 3 | Simple Linear Regression and Correlation Analysis | 15 |

#### M.6.2.1 Section 1

**Section title:** Sampling Distributions Associated with the Normal Population

**Topics covered in this section:**

- Introduction to the course, toward inference
- Student's t-distribution
- Bernoulli and binomial distribution
- Chi-square distribution
- Snedecor's F-distribution

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Deduce the probability mass function $P(X \leq k$ for a binomial distribution?
2. Let X1,...,Xk be $k$ iid random variables distributed with a $\chi^2$ distribution with n1,...nk degrees of freedom respectively.
   What is the distribution of Y=X1+...+Xk? Define it precisely and prove the answer formally?
3. List at least 3 random variables that "tend to follow" a t distribution?
4. If X has Chi square function with the 5 degrees of freedom, then what is the probability that X is between 1.145 and 12.83?
5. If X has a gamma distribution of (1,1), then what is the probability density function of the random variable 2X?

**Typical questions for seminar classes (labs) within this section**

1. Define and provide examples of sample space, events and probability measure.
2. Write the formula for the coefficients of the simple linear regression. Explain the mathematical procedure you do to derive them and derive them.
3. Calculate the correlation between two functions and explain its meaning.
4. Calculate the Pearson coefficient for the given functions.

5. Deduce the MGF for normal distribution.
6. State and prove the Bonferroni inequality.

**Test questions for final assessment in this section**

### M.6.2.2 Section 2

**Section title:** Test of Statistical Hypotheses

**Topics covered in this section:**

- Z-test
- Student's t-test
- Chi-square test
- Snedecor's F-test

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Define the concept of power of a statistical test.
2. Define the purpose of the F Test, its hypotheses, and its structure.
3. Define the purpose of the t-Test, its hypotheses, and its structure.
4. Define the purpose of the Chi square Test, its hypotheses, and its structure.
5. Define the purpose of the Z Test, its hypotheses, and its structure.
6. Provide concrete numeric examples with explanation on why the power of a test depends on:
   (a) the size of the data sets.
   (b) the magnitude of the effect.
   (c) the level of statistical significance.

7. Given a statistical test for which we have set a value $\alpha$ we obtain a p:
   (a) if we can reject H0 ($p < \alpha$), what we typically say about H0 and H1.
   (b) if we cannot reject H0 ($P \geq \alpha$), what we can typically say about H0 and H1.
   (c) when can we say that H0 holds?
   (d) when can we say that H1 holds?

**Typical questions for seminar classes (labs) within this section**

1. Provide a concrete example of a t test, detailing both H0 and H1.
2. Present the structure of the F test for the analysis of the variance.
3. Explain what are H0 and H1 in hypothesis testing.
4. Explain the role of the Bonferroni inequality in hypothesis testing.

**Test questions for final assessment in this section**

#### M.6.2.3 Section 3

**Section title:** Simple Linear Regression and Correlation Analysis

**Topics covered in this section:**

- Kolmogorov-Smirnov test
- Size of samples, Kolmogorov-Smirnov, Fisher exact
- Logistic regression

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Let X1,X2, ...,X10 be a random sample from a distribution whose probability density function is $f(x) = (1 if 0 < x < 1$, otherwise 0). Based on the observed

values 0.62, 0.36, 0.23, 0.76, 0.65, 0.09, 0.55, 0.26, 0.38, 0.24, test the hypothesis H0 : X UNIF(0, 1) against H1 : X UNIF(0, 1) at a significance level = 0.1.

2. If X1,X2, ...,Xn is a random sample from a distribution with density function $f(x) = ((1-\theta)x^{\theta} if 0 < x < 1$, otherwise 0), what is the maximum likelihood estimator of $\theta$?
3. Let X1,X2, ...,Xn be a random sample of size n from a distribution with a probability density function $f(x) = ((1-\theta)x^{\theta} if 0 < x < 1$, otherwise 0), where $0 < \theta$ is a parameter. Using the maximum likelihood method find an estimator for the parameter $\theta$.
4. Suppose you are told that the likelihood of $\theta$ at $\theta = 2$ is given by $1/4$. Is this the probability that $\theta = 2$? Explain why or why not.

**Typical questions for seminar classes (labs) within this section**

1. If X1,X2, ...,Xn is a random sample from a distribution with density function $f(x) = (\frac{1}{\theta} if 0 < x < 1$, otherwise 0), then what is the maximum likelihood estimator of $\theta$?
2. Let X1,X2, ...,Xn be a random sample from a normal population with mean $\mu$ and variance $\sigma^2$. What are the maximum likelihood estimators of $\mu$ and $\sigma^2$?
3. Suppose that you have the following data points: 0.36, 0.32, 0.10, 0.13, 0.45, 0.11, 0.12, 0.09; compute Dn to determine if they come from the uniform distribution [0,0.5].
4. The data on the heights of 12 infants are given below: 18.2, 21.4, 22.6, 17.4, 17.6, 16.7, 17.1, 21.4, 20.1, 17.9, 16.8, 23.1. Test the hypothesis that the data came from some normal population at a significance level = 0.1.

### M.6.2.4 Test questions for final assessment in the course

1. Providing full example of two sequences (in case of computational overhead, you can approximate at the first decimal digit). Compute their:
   (a) Covariance.
   (b) Pearson's correlation coefficient.
   (c) Spearman's Rank Correlation Coefficient.
   (d) Kendall's tau Correlation coefficient.

2. What is an empirical distribution?
3. Present, prove, and discuss the evaluation of the asymptotic confidence interval for the empirical distribution, detailing the role of the binomial.
4. Prove, under the simplified hypotheses, the distribution free property of Dn.
5. Write the Shannon Theorem and discuss its implications.
6. Discuss how we could proceed to compute the confidence interval of the Kendall Tau correlation coefficient of the population.
7. Suppose that you have the following datapoints: 0.4, 2, 0.6, 2.4, 2.2, 3.6, 3.8, 4; compute Dn to determine if they come from the uniform distribution [0,4].
8. Prove that $\tilde{F}_n$ is a consistent and unbiased estimator of F.

## M.7 Analysis of Software Artifacts

- **Course name:** Analysis of Software Artifacts
- **Course number:** SE-01

### M.7.1 Course characteristics

#### M.7.1.1 Key concepts of the class

- Quality Models and Metrics
- Technical Debt
- Verification Techniques including Testing, Static Analysis and Inspection
- Adequacy Criteria
- Process Quality

#### M.7.1.2 What is the purpose of this course?

Software quality is a key aspect of any IT solution whether a few hundred lines of code for a smart phone app or a few million lines of code for Enterprise Resource Planning software. The Analysis of Software Artifacts course provides techniques to develop confidence in the quality of the software being produced or acquired regardless of its size and domain. The course adopts the view that software quality is not only the absence of defects but that it encompasses all the characteristics that bear on the ability of the software to satisfy stated and implied needs. Software quality is then defined from different perspectives: product quality, quality in use and process quality through the use of specific quality models. The course systematically explores different quality attributes and the techniques most appropriate to verify them. Specific topics include software testing, static analysis and model checking, inspections, technical debt, cost of software quality, planning for quality, quantitative models and defect classifications. The course balances traditional lectures with small projects in which students apply the ideas they are learning to real artifacts. The final project consists on the preparation of a quality plan for an industry project.

#### M.7.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students will remember:

- Several views on software quality.
- Trade-offs among quality attributes in quality models.
- Major differences verification techniques.
- Adequacy criteria for verification.
- Cost of quality.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- Quality models usage.
- Technical debt concept.
- Strengths and weaknesses of specific verification techniques.
- Quality planning.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Define and execute a testing plan.
- Perform static analysis of the code.
- Define a quality plan.
- Justify quality related decisions to different stakeholders.

#### M.7.1.4 Course evaluation

### M.7.2 Evaluation

The students performance will be evaluated as follows:

- Mid-term exam (20%)

Table M.18: Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 10 |
| Interim performance assessment | 30 | 50 |
| Exams | 50 | 40 |

- Final exam (20%)
- Quality plan (10%)
- Group projects (20%)
- Individual Assignments (20%)
- Participation (10%)

#### M.7.2.1 Grades range

Table M.19: Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 65-79 |
| C. Satisfactory | 60-74 | 50-64 |
| D. Poor | 0-59 | 0-49 |

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table M.73, but it may change slightly (usually reduced) depending on how the semester progresses.

#### M.7.2.2 Resources and reference material

- Text book:
- This course makes use of many reference materials that are posted to Moodle:
- David A. Garvin, What Does "Product Quality" Really Mean?
- Volker Kruger, Main Schools Of Tqm: "the big five"
- Steve McConnell, Managing Technical Debt
- Jean-Louis Letouzey, Michel Ilkiewicz, Managing Technical Debt with SQALE Method

- Stephen Chin, Erik Huddleston, Walter Bodwell, and Israel Gat, The Economics of Technical Debt
- Douglas W. Hubbard, How to Measure Anything. Finding the Value of "Intangibles" in Business
- SEI, Foundations of Measurement
- Frank Buechner, Is 100% Code Coverage Enough?
- Brian Marick, How to Misuse Code Coverage
- Ben H. Smith, Laurie Williams, Should software testers use mutation analysis to augment a test set?
- Frank Buechner, Test Case Design Using the Classification Tree Method
- Thomas J. McCabe, Arthur H. Watson, Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric
- Richard Hamlet, Random Testing
- Matt Warnock, Look out! It's the fuzz!
- Karl E. Wiegers, Peer Reviews in Software: A Practical Guide
- DoD, Formal Inspections
- Jason Cohen, Questions for a Review Process
- Cohen, Prepare to Succeed - A Guide to Effective Code Review
- Michael A. Howard A Process for Performing Security Code Reviews
- Stephan Wagner, Software Quality Economics for Combining Defect-Detection Techniques, 2005
- Ashok Shenvi, Defect Prevention with Orthogonal Defect Classification, 2009
- Brian Chess, Jacob West, Secure Programming with Static Analysis, 2007
- Mordechai Ben-Ari, A Primer on Model Checking, 2010
- Dave Jewell, Performance Engineering and Management Method, 2008
- Jane Hillston, Performance Modeling, Operational Laws
- Craig Shallahamer, Forecasting Oracle Performance (Ch. 5, Practical Queuing Theory), 2007
- Gerald Everett, Performance Testing, Chapter 9
- IEEE Guide for Software Verification and Validation Plans, 1993
- Rick D. Craig, Systematic Software Testing, 2002
- Peter Mell, A Complete Guide to the Common Vulnerability Scoring System, 2007

- NIST, Technical Guide to Information Security Testing and Assessment, 2008
- Boris Mutafelija, Systematic Process Improvement Using ISO 9001:2000 and CMMI, 2003
- Edward F. Weller, Practical Applications of Statistical Process Control, 2000
- Larry Webber, Michael Wallace, Quality Control for Dummies, 2007
- Mahesh S. Raisinghani, Six Sigma: concepts, tools, and applications, 2005
- SEI, Practical Software Measurement: Measuring for Process Management and Improvement, 1997

## M.7.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.20:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Defining quality | 6 |
| 2 | Testing | 16 |
| 3 | Static Analysis | 8 |
| 4 | Advanced Analysis and Verification | 10 |
| 5 | Quality planning | 14 |

### M.7.3.1 Section 1

**Section title:**

Defining quality

**Topics covered in this section:**

- Introduction, Views on Quality
- Quality Models
- Measurements & Quality Metrics

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. What is the dominant quality view implicit in SCRUM and RUP?
2. Explain in your own words and in no more than three sentences the main contribution of one of the quality gurus like Ishikawa?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

3. What is the difference between must have attributes and delighters in Kano's concept?
4. What is the main difference between a quality model like ISO 25010 and SAP Products Standard?
5. Describe in your own words and with regards to ISO 25010 the following quality attributes: Security, Reliability and Maintainability.

**Typical questions for seminar classes (labs) within this section**

1. Define major quality focus by the customer in a given project.
2. Using SONAR evaluate maintainability of a given project.
3. Discuss you interpretation of the obtained quality level in a given project.
4. Describe how and what for quality models are useful? Provide an example from your studio project.
5. Map the requirement "the system shall be easy to maintain" to the ISO 25010 Quality model. Provide a definition to the metric level for at least two sub-characteristics for the requirement, and represent the mapping graphically.

**Test questions for final assessment in this section**

1. Explain the difference between product quality, quality in use and process quality. Provide 2-3 quality attributes of each category briefly describing them.
2. What quality view best encompasses the phrase "Quality consists of the extent to which a specimen [a product-brand-model-seller combination] possesses the service characteristics you desire".
3. Explain the difference between accuracy and precision of measurement methods.
4. For each of the following quantities, indicate the scale (nominal, ordinal, interval,

or ratio) of the data (just the scale, no justification required): a. Categories of defect types in a bug database. b. Branch coverage of a test suite. c. Severity of the defects in a bug database. d. Statement coverage of a test suite. e. Number of features delivered on a milestone.

#### M.7.3.2 Section 2

**Section title:**

Testing

**Topics covered in this section:**

- Verification Overview
- Measuring Test Adequacy
- Black Box Testing
- Modeling the Input Domain
- Combinatorial Testing
- Basis Path & Data Flow Testing
- Random & Mutation Testing

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. In the context of mutation testing: a. What is an equivalent mutant? b. What is the meaning of the terms killed and dead on arrival? c. What is the difference between the two?
2. Develop BVA test cases for an application that implements the logic as defined in the exercise.
3. Will you use combinatorial testing to derive test cases for a tree like menu? Yes, no, why?

4. What is the relation between branch coverage and mutation testing?
5. What is an infeasible path?
6. What is fuzz testing? How it is different from random testing?
7. What is the oracle problem?

**Typical questions for seminar classes (labs) within this section**

1. Write a short code snippet that contains a possible null-pointer exception, and two different sets of test cases that achieve full branch coverage for the snippet. The first set of test cases should miss the defect; the second should trigger it.
2. Develop a classification tree covering all test relevant aspect for a Merge method. The method accepts two ordered integer vectors with a maximum of 128 elements each and returns a single ordered vector with no-duplicates formed from the elements of the input vectors.
3. Develop test cases for the logical function (A & B) | C -> D so that it achieves 100% MC/DC.
4. Develop test cases to achieve 100% basis path coverage utilizing McCabe method for the program below. Include: control flow graph, basis paths, test cases.

**Test questions for final assessment in this section**

1. Identify equivalence classes using Decision Table Method for a given problem.
2. Create a Classification Tree for the Problem. Identify constraints. Indicate boundaries.
3. Calculate number of test cases to achieve Basis Path coverage for a code sample.
4. Provide a test set that achieves full Basis Path coverage for a code sample.

#### M.7.3.3 Section 3

**Section title:**

Static Analysis

**Topics covered in this section:**

- Inspections

- Static Analysis
- Model Checking

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Wiegers references Weinberg's concept of "egoless programming." What does he mean by this concept, and why is it relevant to peer review?
2. The literature suggests a number of limits on code review sessions. For each, list and justify a reasonable guideline.
3. Based on your reading, list two undesirable programmer attitudes that can emerge in an organization that includes mandatory code reviews. Describe three mechanisms management, organization, or programmers can use to avoid the development of such attitudes.
4. Under what circumstances is model checking not a useful strategy?
5. In the context of model checking, what is a counterexample?
6. What is one common misconception about either the advantages or disadvantages of model checking versus static analysis that Engler and Musuvathi identify and debunk through experience in their article "Static analysis versus software model checking for bug finding"?

**Typical questions for seminar classes (labs) within this section**

1. Using Humphrey's capture-recapture procedure, how many latent defects can we estimate remain unidentified for a given code?
2. You need to inspect a large banking system comprising around 20,000 lines of COBOL and 25,000 lines of newly written Java code with regards to vulnerabilities but only have enough budget to look at 10,000. How would you prioritize which components to inspect without overrunning the budget?

3. Produce a program model for a given code example. Be sure to identify and describe the states, actions, transitions, initial state, and end states.
4. Create a Promela model that will print out even integers from 0 to 100.

**Test questions for final assessment in this section**

1. Enumerate three limitations of many dynamic analyses, and describe a mitigation strategy to overcome each.
2. Give an example of a circumstance (in terms of a system, property, program, defect, or pattern type) under which you would prefer: a) Dynamic over static analysis? b) Static over dynamic analysis c) Model checking over more lightweight static analysis?
3. Describe two strategies for eliciting developer support and encouraging analysis tool adoption in an organization.
4. Define the terms "sound" and "complete" with respect to an analysis tool, and explain or give examples of circumstances under which you would prefer one over the other in selecting a particular tool.

#### M.7.3.4 Section 4

**Section title:**

Advanced Analysis and Verification

**Topics covered in this section:**

- Performance Analysis & Verification
- Maintainability Analysis & Verification
- Security Analysis & Verification
- Organizational Quality & Process Improvement

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Explain impact on Utilization on Response Time.
2. Explain Amdahl's law relation to performance improvements during development process.

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

3. Explain Little's law application to evaluation of critical performance characteristics of systems (response time, queue size...)
4. Give definition of maintainability. What are the statements of Lehman's Laws on Program Evolution?
5. Give an example of vulnerability prevention measures?
6. What is Juran's view on process improvement? Which CMMI level will best suite Juran's methods.

**Typical questions for seminar classes (labs) within this section**

1. You execute a benchmark test twice and find that the performance of the system was 30 transactions/hour the first time and 20 transactions/hour the second time. What is the average throughput?
2. From a pure performance point of view is it better, the same or worst to have a single server or two with half the speed?
3. You execute a load test for one hour, first during peak hour and again off-peak. During peak hour the system process 20 transactions/hour. Off-peak it processes 30 transactions/hour. What is the average throughput?
4. Your current e-commerce traffic, 12.5 transactions per second, is served by two CPUs running at 65% of its maximum capability. With the launch of a new product, marketing is forecasting an increase of 30% in your web site traffic. Your job is to make a recommendation, from a pure performance perspective, on whether to upgrade your current system with faster CPUs or to buy two additional CPUs with the same capacity as the existing ones.It is estimated the faster CPUs will reduce the current service time by 20%.
5. Draw a queuing diagram for the systems below and describe them using Kendall's a) Single CPU system. b) A system comprising three web servers, to which re-

quests are randomly directed. Each of the servers contains two CPUs.

6. Software monitor data for an interactive system shows a CPU utilization of 75%, a 3 second CPU service demand, a response time of 15 seconds, and 10 active users. What is the average think time of these users?
7. Construct a Design Structure Matrix for a given set of components. What does the DSM analysis tell you about maintainability of this set of components?

**Test questions for final assessment in this section**

1. Give an example illustrating general relationships between response time, throughput, and resource utilization.
2. Suppose that during an observation period of 1 minute, a single resource (e.g., the CPU) is observed to be busy for 36 sec. A total of 1800 transactions were observed to arrive to the system. The total number of observed completions is 1800 transactions (i.e., as many completions as arrivals occurred in the observation period). What is: a) The mean service time per transaction, b) The utilization of the resource, c) The system throughput?
3. Given a table listing arrival time of groups of new transactions and transactions currently in system, calculate a response time of the system.
4. A web server is monitored for 10 minutes and its CPU is observed to be busy 90% of the time. The web server log shows that 30,000 requests were processed in that period. What is the CPU service demand of the web server?
5. Give an example of the effect of the law of increasing complexity on maintainability.
6. Give an example of fuzzing for security testing.
7. Give a brief definition of CMMI levels.

#### M.7.3.5 Section 5

**Section title:**

Quality Planning

**Topics covered in this section:**

- Technical Debt

- Quality Planning - Cost of Quality
- Quality Planning - Project Quality
- Quality Plan for Practicum Project

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is Kruchten's definition and taxonomie of Technical Debt?
2. According to Highsmith, what is relation of Technical Debt and Cost of Change?
3. In McConnell's taxonomy which type of Technical Debt can be positive?
4. Explain latent faults through "tank and pipes" model. Give an example.
5. What is the Quality Plan? Give an example of an estimation method for the efforts required to implement the quality plan.

**Typical questions for seminar classes (labs) within this section**

1. Give definition of quality artifacts contributing to the SQALE model.
2. Based on the experience with the group project, do the calculated Technical Debt metrics correspond to your intuition? Justify.
3. Give an example of possible appraisal costs for a given project.
4. Present the quality model for the practicum project.
5. Present the quality control measures for the practicum project.
6. Present the quality plan for the practicum project with regards to the project milestones and available resources.

**Test questions for final assessment in this section**

1. Explain the different types of technical debt that a project might incur.
2. Give a definition of constituent parts of the cost of quality.

3. Given project characteristics, what quality attributes should be tracked throughout of the project. Give an example of a quality control measures for the top priority quality attribute.
4. What are the quality gates? Give an example for quality gates at 2 different milestones.

# M.8 Architectures of Software Systems

- **Course name:** Architectures of Software Systems
- **Course number:** SE-02

## M.8.1 Course characteristics

### M.8.1.1 Key concepts of the class

- Architectural styles and patterns
- Tactics
- Quality attribute scenario
- Architectural documentation
- Architecture tradeoff analysis method

### M.8.1.2 What is the purpose of this course

Modern software systems are complex and their development require a deep knowledge of how to control the growing complexity starting from correct choices in the architectural design. The Course is a reflection on the comprehensive approach to modern software architectural design to stimulate the SW engineers sensitivity to the robustness and the resilience of the applications. Software engineering and software architecture concepts will be deconstructed and reconstructed considering well-known and innovative approaches used in the real world to manufacture software solutions through lectures and practical activities. The students will develop real applications on the subject of project courses assigned to different competing teams. Individual contributions will be evaluated as well as the participation in teamwork.

### M.8.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students will remember:

- A number of architectural patterns.
- Quality attributes and tactics to promote them.

- Different perspectives for documenting software architecture.
- Module, component-and-connector and allocation categories of styles.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- Dataflow architectural style, patterns belonging to this style, their properties, qualities promoted and inhibited.
- Call-return architectural style, patterns belonging to this style, their properties, qualities promoted and inhibited.
- EVent-based architectural style, patterns belonging to this style, their properties, qualities promoted and inhibited.
- Data-centered architectural style, patterns belonging to this style, their properties, qualities promoted and inhibited.
- Architecture evaluation techniques such as ATAM and ARB
- Static, dynamic and physical perspectives of architecture documentation

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Perform quality attribute scenario analysis.
- Derive architectural drivers from the given requirements and constraints.
- Perform attribute driven architecture design.
- Recognize different architectural patterns.
- Read and design UML diagrams.

### M.8.2 Evaluation

The students performance will be evaluated as follows:

- RQs (30%)
- Assignments (30%)
- Tests (20%)
- Project (20%)

Table M.21: Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 20 |
| Interim performance assessment | 30 | 60 |
| Exams | 50 | 20 |

#### M.8.2.1 Grades range

Table M.22: Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 90-100 |
| B. Good | 75-89 | 75-89 |
| C. Satisfactory | 60-74 | 60-74 |
| D. Poor | 0-59 | 0-59 |

#### M.8.2.2 Resources and reference material

- David Garlan, Felix Bachmann, James Ivers, Judith Stafford, Len Bass, Paul Clements, and Paulo Merson. 2010. Documenting Software Architectures: Views and Beyond (2nd ed.). Addison-Wesley Professional.
- Anthony J. Lattanze. 2008. Architecting Software Intensive Systems: A Practitioners Guide (1st ed.). Auerbach Publications, Boston, MA, USA.
- Len Bass, Paul Clements, and Rick Kazman. 2012. Software Architecture in Practice (3rd ed.). Addison-Wesley Professional.

### M.8.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

Table M.23: Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Structures and Properties | 6 |
| 2 | Documenting Designs | 4 |
| 3 | Architecting for X | 14 |
| 4 | Styles and Patterns | 22 |

#### M.8.3.1 Section 1

**Section title:** Structures and Properties

**Topics covered in this section:**

- Software architecture definition
- Architectural drivers
- Design structures

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is a view and how is it different from a structure?
2. What is "perspective" and how does it affect our analysis of structures?
3. What are architectural drivers?
4. What is system architecture?
5. What is the role of architect?

**Typical questions for seminar classes (labs) within this section**

1. Given the following requirement: "The system shall have the capability to detect software faults and automatically restart the corresponding component." Are there potential problems with this requirement - does this seem like a reasonable requirement or not? If not, describe the problems that with this requirement.
2. Decompose given requirement according to quality attribute scenario.
3. Explain the difference between enterprise, solution and system architect.

#### M.8.3.2 Section 2

**Section title:** Documenting Designs

**Topics covered in this section:**

- Architecture documentation views and perspectives
- Documentation techniques
- Notations for documentation

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Is architectural documentation relevant for systems developed using agile methods or for systems whose architecture changes frequently?
2. What perspectives are intended to be represented by each documentation style?
3. Write the general structure of the OLS equation for one variable.
4. What is a sequence diagram?
5. What are the different interaction diagram notations does UML have?

**Typical questions for seminar classes (labs) within this section**

1. Explain briefly how architectural documentation needs to be adapted to the context of agile development.
2. List three kinds of stakeholders and identify specifically the kind of information they would need to extract from the document. What kind of views would likely contain this information?

### M.8.3.3 Section 3

**Section title:** Architecting for X

**Topics covered in this section:**

- Architecting for Performance, Availability, Security

- ADD technique
- Guidance for the Architect
- Architecture Evaluation
- Architecture and Design

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. What is real-time scheduling and what are the key scheduling issues?
2. What is the difference between model-based analysis and measurement-based analysis for analyzing reliability and availability?
3. How does Least Privileges principle promote security in a system?
4. what do you think is an important strength and weakness of the ATAM?
5. What issues and observations can ACDM help design teams address?

**Typical questions for seminar classes (labs) within this section**

1. What approach would you use to analyze the availability of a data server – why?
2. How might promoting availability in a system impact change-oriented quality attributes
3. Describe what a fault model is and what it's used for.
4. Give 3 possible response mechanisms once a fault has been detected.
5. Describe the types of intrusion indicators.
6. Contrasting ATAM with the Architecture Review Board process describe 3 differences between the two review processes

#### M.8.3.4 Section 3

**Section title:** Styles and Patterns

**Topics covered in this section:**

- Dataflow Styles
- Call-Return Styles
- Client-Server and Tiered Architectures
- Middleware
- Event Styles
- Data-centered styles

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. With respect to layered structures: (a) what are the elements and relations in a layered structure?
2. Under what conditions would client server provide better performance than a 3-tiered system and conversely, under what conditions would a 3-tiered system provide better performance than client server?
3. What is heterogeneity and what challenges does it present?
4. What role does middleware play in managing and resolving heterogeneity?
5. What are the key challenges in wireless sensor networks and cyber physical systems that are not experienced in large scale development?
6. What is the basic structure of the Relational Database Model and what does this model hide?
7. What is the difference between and analytical and operational database?
8. What are the essential elements of the Blackboard Pattern?

**Typical questions for seminar classes (labs) within this section**

1. Characterize the pipe-and-filter pattern in terms of perspective, topological arrangement/elements and relationships, semantics, qualities promoted and inhibited
2. Using only call-return components and connectors, how could you mimic a pipe?
3. Explain what middleware is, why we use it, and the general categories of middleware
4. Describe what DDS is and summarize its unique strengths in comparison to similar middleware technologies.
5. Describe a situation where an event-oriented pattern might be a better choice than call-return and why.
6. Describe a situation where call-return pattern might be a better choice than event-oriented and why.
7. What is a NoSQL database and describe the three general data representation models.
8. Describe the functional role each element plays in the Blackboard pattern.

#### M.8.3.5 Test questions for final assessment in the course

1. Is it possible to get rid off from FR and describe the system by Quality Attributes and Constraints only?
2. You got all the required information from stakeholders about the business goals that system should achieve and impact of each of them (business goals) to the system (system's architecture). Is it true that you have everything for preparing list of ASRs?
3. What groups participated in ATAM are external to the project?
4. What are the perspectives Call-Return patterns might be depicted from?
5. Which role in ATAM (Architecture Tradeoff Analysis Method) is best described as follows: Sets up the evaluation; establishes evaluation contract; forms evaluation team; sees that final report is produced and delivered
6. What pattern is more appropriate in developing a system that changed, added or removed process data?
7. In six part Quality Attribute Scenario, what part can be described as "a condition

that requires a response when it arrives at a system"?

8. What is Software Architecture?
9. To which category of security tactics Ignore Faulty Behavior belongs?
10. What quality attribute does Single Access Point pattern promote?
11. What is the right sequence of steps for implementing ADD?
12. Which patterns belong to the Dataflow style?
13. What quality attribute (according to the ISO/IEC 25010 quality model) is disrupted by a successful denial of service attack?
14. What is Component-and-Connector View is used for?

## M.9 Behavioral and cognitive robotics

- **Course name:** Behavioral and Cognitive Robotics
- **Course number:**

### M.9.1 Course Characteristics

### M.9.2 What subject area does your course (discipline) belong to?

Robotic control; Artificial Intelligence; Cognitive Science; Machine Learning.

#### M.9.2.1 Key concepts of the class

- Behavior and cognition as complex dynamical systems
- Embodied cognition
- Machine learning methods for robotics

#### M.9.2.2 What is the purpose of this course?

The course illustrates the fundamental principles of robotics with particular reference to autonomous robotics: embodiment and situatedness, morphological computation, sensory-motor coordination, robustness, emergence, multi-levels and multi-scale organizations, compositionality, integration between behavioral and cognitive skill, anticipation, and internal models. Students will learn how to program robots and how to design robots that develop their behavioral and cognitive skills autonomously, in interaction with the environment, through evolutionary and reinforcement learning algorithms. This will be realized through the illustration of concrete examples taken from state of the art research and through practical experimentation carried in simulation.

#### M.9.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize

- What makes situated and embodied system different from other AI applications

- The importance of sensory-motor coordination and morphological computation
- The advantage and drawbacks of adaptive methods and of methods based on hand-design
- The open research issues in autonomous robotics

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- The importance of the interaction between the body of the robot, the controller of the robot and the environment
- The multi-level and multi-scale organzation of behavior and cognition
- The characteristics of evolutionary, reinforcement learning, and learning by demonstration methods
- The relation of the methods above with supervised learning methods
- The methods that can support long-term adaptation and the synthesis of complex solutions

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Design a controller for a mobile robot capable to perform a simple task
- Evolve a robot for the ability to perform a task by using evolutionary algorithm
- Train a robot for the ability to perform a task by using reinforcement learning algorithm
- Understand applicability of these methods for other sequential decision problems (e.g. recomendation systems, chatbots, educational technologies ext)

#### M.9.2.4 Course evaluation

The course grades are given according to the following rules: Labs/assignments = 25 pts, Interim project report = 25 pts, Final project report = 50 pts.

**Table M.24:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/assignments | 25 | 25 |
| Interim project report | 25 | 25 |
| Final project report | 50 | 50 |

#### M.9.2.5 Grades range

If necessary, please indicate freely your course's grading features.

**Table M.25:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

#### M.9.2.6 Resources and reference material

Annotated slides, resources available from https://github.com/snolfi/evorobotpy

### M.9.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.26:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction to autonomous robots | 8 |
| 2 | Behavior | 7 |
| 3 | Cognition | 7 |

#### M.9.3.1 Section 1

**Section title:** Introdution to autonomous robots

**Topics covered in this section:**

- Autonomous robots
- From Braitenberg's vehicles to neuroRobots
- Embodiment
- Situatedness
- Adaptation
- Robustness, Plasticity, and Antifragility

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Identify the role of sensory-motor coordination in the behavior exhibited by a specific robot
2. Describe an example of morfological computation
3. Describe the differences between evolutionary and reinforcement learning methods
4. Describe the criteria that can be used to design reward functions

**Typical questions for seminar classes (labs) within this section**

1. Implement a simple evolutionary algorithm in python
2. Familiarize with the OpenAI Gym environment
3. Evolve a robot by using the evorobotpy tool
4. Train by using a reinforcement learning algorithm included in the baseline tool

**Test questions for final assessment in this section**

1. Use the evolutionary algorithm implemented to evolve a solution for pole-balancing problem

2. Analyze the impact of an hyper-parameter on the course of the learning process
3. Analyze the impact of an atchitecture of the policy on the course of the learning process

#### M.9.3.2 Section 2

**Section title:** Behavior

**Topics covered in this section:**

- Behavior as a complex adaptive system
- Swarm robotics
- Coordination, communication and self-organization in collected behavior
- Behavior arbitration and affordances

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Describe the role of positive and negative feedbacks in self-organized behaviors
2. Describe the different mechanisms supporting behavior arbitration
3. Describe examples of fixed point, limit cycle, and strange attractor behaviors

**Typical questions for seminar classes (labs) within this section**

1. Implement a new Gym environmental problem
2. Analyze the impact of alternative reward functions
3. Design a reward function

**Test questions for final assessment in this section**

1. Train a robot for the ability to perform multiple behaviors

2. Design an experimental setup supporting the evolution of cooperation and specialization
3. Implement a procedure capable to self-select usefull learning experience

#### M.9.3.3 Section 3

**Section title:** Cognition

**Topics covered in this section:**

- Symbols and Language
- Prerequisites for the development of cognitive skills
- Integration between behavioral and cognitive skills
- Learning by demonstration
- Internal Models
- Long-term adaptation

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Describe the advantage and the limits of sensory-motor coordination
2. Describe the processes supporting the evolution of signals and meanings in communication
3. Illustrate the symbol-grounding problem

**Typical questions for seminar classes (labs) within this section**

1. Compare the solutions synthesized through reactive and recurrent neural policy
2. Analyze the role played by sensory fusion
3. Analyze the importance of normalization techniques in reinforcement learning

**Test questions for final assessment in this section**

1. Implement an auto-associative network to extract compressed visual features
2. Implement an anticipatory models that permit to handle the temporal lack of sensory information
3. Compare the efficacy of vanilla and regulatory neural network policy

# M.10 Big Data Technologies and Analytics

- **Course name:** Big Data Technologies and Analytics
- **Course number:** N/A

## M.10.1 Course Characteristics

### M.10.1.1 Key concepts of the class

- Advanced distributed data organization
- Advanced distributed data processing

### M.10.1.2 What is the purpose of this course?

Nowadays companies need to manage vast amounts of data on a daily basis. Storing, sorting, accessing and analyzing obtaining synthetic information is considered one of the great challenges of the 21st century and and being effective in this may make the difference between success and failure. In order to gain a competitive advantage, Big Data and Analytics professionals are able to extract useful information from data and increase the Return Of Investments. In this course, students will be exposed to the key technologies and techniques, including R and Apache Spark, in order to analyze large-scale data sets and uncover valuable business information.

### M.10.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

- Understanding of big data applications.
- Algorithms for the statistical analysis of big data
- Fundamental principles of predictive analytics

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to ...

- How to process batch data

- How to process stream data
- Advanced design of distributed architectures
- Advanced design of distributed algorithms

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

- Write a program for batch processing
- Write a program for stream processing
- Design distributed processing pipelines
- Desing distributed algorithms

### M.10.1.4 Course evaluation

**Table M.27:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 30 |
| Interim performance assessment | 30 | 30 |
| Exams | 50 | 40 |

If necessary, please indicate freely your course's features in terms of students' performance assessment.

### M.10.1.5 Grades range

**Table M.28:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

If necessary, please indicate freely your course's grading features.

#### M.10.1.6 Resources and reference material

- Slides and material provided during the course.
- F. Provost and T. Fawcett. Data Science for Business. O'Reilly, 2013
- Matthew North. Data Mining for the Masses, Second Edition: with implementations in RapidMiner and R. CreateSpace Independent Publishing Platform, 2012
- Tom White. Hadoop: The Definitive Guide. O'Reilly Media, Inc., 2012
- Seema Acharya and Subhashini Chellappan. Big data and analytics. WileyIndia, 2016

### M.10.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.29:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction | 2 |
| 2 | File systems and resource managers | 8 |
| 3 | Batch Processing | 8 |
| 4 | Stream Processing | 8 |
| 5 | Analytics | 4 |

#### M.10.2.1 Section 1

**Section title:** Introduction

**Topics covered in this section:**

- What is Big Data
- Characteristics of Big Data
- Technologies
- Virtualization and cloud computing

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Describe the 6 Vs
2. Describe the technologies to support big data

**Typical questions for seminar classes (labs) within this section**

1. Design the structure of a cloud architecture for big data
2. Give examples of the 6 Vs in real systems

**Test questions for final assessment in this section**

1. Design the structure of a cloud architecture for a specific analytics type
2. Give examples of the 6 Vs in real systems

#### M.10.2.2 Section 2

**Section title:** File systems and resource managers

**Topics covered in this section:**

- HDFS
- YARN

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Describe the characteristics of the different nodes of HDFS
2. How files and blocks are managed
3. Describe the resource manager
4. Describe the lifecycle of an application
5. Describe and compare the scheduling approaches

**Typical questions for seminar classes (labs) within this section**

1. Configure a HDFS cluster
2. Build a HDFS client
3. Use a HDFS command line
4. Configure YARN
5. Evaluate the overall performance of YARN

**Test questions for final assessment in this section**

1. Configure a HDFS cluster with some specific replication approaches
2. Build a HDFS client
3. Evaluate the performance of a specific configuration
4. Compare the different schedules

### M.10.2.3 Section 3

**Section title:** Batch Processing

**Topics covered in this section:**

- Distributed batch processing
- MapReduce model
- Applications
- Tasks management
- Patterns

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Describe the MapReduce model
2. Describe tasks management
3. Describe patterns of usage

**Typical questions for seminar classes (labs) within this section**

1. Solve with MapReduce a specific problem
2. Implement a usage pattern

**Test questions for final assessment in this section**

1. Describe the advantages and disadvantages of the MapReduce model
2. Solve a task designing the solution using MapReduce
3. Solve a task designing the solution using a composition of usage patterns

#### M.10.2.4 Section 4

**Section title:** Stream Processing

**Topics covered in this section:**

- CAP theorem
- Distributed storage and computation
- Distributed Stream Processing
- Usage patterns

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Analyze the CAP theorem
2. Define the kinds of data storage available
3. Characteristics of stream processing
4. Describe the usage patterns

**Typical questions for seminar classes (labs) within this section**

1. Build a program to solve a problem with stream processing
2. Interact with a NoSQL database

**Test questions for final assessment in this section**

1. Identify problems and solutions related to the CAP theorem
2. Compare solutions with batch and stream processing approaches
3. Design a system using a NoSQL database

### M.10.2.5 Section 5

**Section title:** Analytics

**Topics covered in this section:**

- Architecture
- Use cases
- SparkML
- GraphX

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Features of SparkML
2. Features of GraphX

**Typical questions for seminar classes (labs) within this section**

1. Write a program using SparkML
2. Write a program using GraphX

**Test questions for final assessment in this section**

1. Extend the SparkML library with a custom algorithm
2. Extend the GraphX library with a custom algorithm

# M.11 Classical Internet Applications

- **Course name:** Classical Internet Applications (CIA)
- **Course number:** SNE-03

## M.11.1 Course characteristics

### M.11.1.1 Key concepts of the class

- Internet applications
- Implementation, configuration, and security of Internet applications

### M.11.1.2 What is the purpose of this course?

CIA course serves as kick-start for the secure system and network engineering Masters program. Before diving into the depth of the topics, the students must know preliminary concepts related to computer networks services/applications therein. This course is designed to cover the basic services offered by the internet. The concepts from this course will be used throughout the course of whole masters. More precisely, this course will cover the Domain Name Services (DNS), DNSSec, Email, Web, Directories, and Disks. The theory part will strengthen the theoretical aspects of the concepts whereas the lab exercises will provide the students with the opportunity to have hands-on experience of the ideas they learnt in the lectures.

### M.11.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define

- Identify different Internet applications and understand their working principles from protocols point of view
- Demonstrate the acquired knowledge and skills in classical internet applications including DNS, Email, and Directory services.
- Able to write regular expressions and context-free grammar
- Able to partition disks and remember the booting principles

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- Demonstrate knowledge and skills to use web services
- Demonstrate the essential knowledge of disks and calculate particular locations/addresses in disks
- Reason about problems in the current DNS and the need to upgrade to DNSSEC

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply

- Install, Configure, update, and manage DNS services over a network
- Configure, maintain, and update the secure DNS over a network
- Update, add, and delete records in DNS
- Configure a secure mail server and maintain it
- Get hands-on experience of the afore-mentioned technologies on their own servers.

#### M.11.1.4 Course evaluation

**Table M.30:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 0 |
| Lab tasks assessment | 60 | 40 |
| Exams | 50 | 40 |

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

#### M.11.1.5 Grades range

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table M.31, but it may change slightly (usually reduced) depending on how the semester progresses.

**Table M.31:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

#### M.11.1.6 Resources and reference material

- Lecture slides
- RFCs

### M.11.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.32:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Intro, DNS and DNSSEC | 12 |
| 2 | Email | 4 |
| 3 | Booting | 2 |
| 4 | Directory Services, Disks, Web, and Protocols | 12 |
| 5 | Labs | 56 |

#### M.11.2.1 Section 1

**Section title:**

Introduction, History of Internet and Security, DNS and DNSSEC

**Topics covered in this section:**

- Introduction
- History of Internet
- History of security
- DNS
- DNSSEC

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Hows does DNS query get resolved?
2. What is iterative and recursive DNS server?
3. How does wildcard work in DNS?
4. What is zone walking in secure DNS?
5. What is delegation in DNS?
6. What is NSEC and NSEC3 records in DNSSEC?

**Typical questions for seminar classes (labs) within this section**

1. Configure DNS and DNSSEC (with specific tasks)

**Test questions for final assessment in this section**

1. How does DNS query get resolved in DNS?
2. How zones are formed?
3. How delegation works in DNS?
4. How resource records are verified in DNSSEC?
5. What is meant by zone walking and how is it avoided?

### M.11.2.2 Section 2

**Section title:**

Email

**Topics covered in this section:**

- Email architecture

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are different protocols used in email?
2. What are the different agents used in email system?
3. How to avoid spamming in email?
4. How to configure email servers?

**Typical questions for seminar classes (labs) within this section**

1. Configure email server
2. Configure anti-spamming techniques

**Test questions for final assessment in this section**

1. What are different agents and their roles in email architecture?
2. How MX records work?
3. How to configure different anti-spamming policies?

#### M.11.2.3 Section 3

**Section title:**

Booting

**Topics covered in this section:**

- Secure booting

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. What are the different steps in booting?
2. What is secure booting?
3. Explain UEFI booting

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

### Typical questions for seminar classes (labs) within this section

1. Implement secure booting

### Test questions for final assessment in this section

1. Same as above

### M.11.2.4 Section 4

### Section title:

Deep Learning

### Topics covered in this section:

- Directory services
- Disks
- Web
- Protocols

### What forms of evaluation were used to test students' performance in this section?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

### Typical questions for ongoing performance evaluation within this section

1. How directory services are implemented?
2. How permissions are set?

3. How disk partitions are made?
4. How different web protocols work?

**Typical questions for seminar classes (labs) within this section**

1. Implement different web services
2. Configure active directory

**Test questions for final assessment in this section**

1. As above

## M.12 Communication

- **Course name:** Communication
- **Course number:** SE-
- **Area of instruction:** Computer Science and Engineering

### M.12.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 1st year of MSc
- **Semester of instruction:** 1st semester
- **No. of Credits:** 2 ECTS
- **Total workload on average:** 72 hours overall
- **Frontal lecture hours:** 2 hours per week.
- **Frontal tutorial hours:** 0 hours per week
- **Lab hours:** 2 hours per week
- **Individual lab hours:** 2 hours per week
- **Frequency:** weekly throughout the semester
- **Grading mode:** letters: A, B, C, D

### M.12.2 Course outline

This course is designed to help computer science and engineering students improve their technical reading, writing and spoken English communication. Students will read texts analytically, identify and respond to persuasion and argument, and develop skills including skimming and scanning, reading for gist, and reading critically. They will assess the visual, textual, paralinguistic and audience engagement elements of oral presentations, as well as participating in giving them, and asking and fielding audience questions. Coursework will support students in recognising the structural and rhetorical elements of technical genres, as well as constructing a portfolio of sample workplace text-types. The processes of drafting, revising and incorporating feedback, both in small groups and individually, will both model and inculcate editing methods and procedures. Attention will be given to teamwork, meeting behaviour, and minute-taking. Case studies are used to bring the textual and lived dimensions of technical communication together. Student reflection on their own learning experiences is a significant means of consolidating learning gains, over the course of the semester. Ultimately, the course provides opportunities for students to develop effective technical communication skills.

### M.12.3 Expected learning outcomes

By the end of this course, students will be able to:

- Recognize and describe the constituent elements of technical writing;
- Appreciate the roles of ethics, persuasion and teamwork in technical communication;
- Know how to read for information, gist and detail;
- Identify key features of different genres of technical writing;
- Organize, deliver and analyses effective and attractive presentations;
- Plan, prepare, write and edit a business proposal;
- Construct a job application package
- Design visual aids and instructional texts for specific audiences;
- Appraise effective sentences, paragraphs and whole-text cohesion for various technical documents;
- Manage the cross-cultural elements of technical communication.

### M.12.4 Required background knowledge

Students must be familiar with how to access and use the Learning Management System. Students should have a level of English of B2 (upper intermediate) or above.

### M.12.5 Prerequisite courses

There are no prerequisite courses.

### M.12.6 Detailed topics covered in the course

- Team work
- Intercultural communication

- Analytic Summary
- Revising and Editing
- Analysing Presentations
- User Manuals
- Instructional texts
- System Specifications
- Office correspondence
- Skimming and Scanning, Critical Reading
- Negotiating Skills
- Application Letters and CV, Job Interviews
- Conducting Meetings
- Agenda and Minutes
- Case Studies

### M.12.7 Textbook

-
-

### M.12.8 Reference material

-
-
-

### M.12.9 Required computer resources

Laptop

### M.12.10 Evaluation

- Oral presentation and negotiation (20%)
- Analytic Summary (10%)
- Reflections (10%)

- Instructional text (20%)
- Business Proposal (20%)
- Job Application Package (10%)
- Participation (10%)

## M.13 Computational intelligence

- **Course name:** Computational Intelligence
- **Course number:** R-02
- **Area of instruction:** Computer Science and Engineering

### M.13.1 Course Characteristics

### M.13.2 What subject area does your course (discipline) belong to?

- Machine Learning
- Optimization

#### M.13.2.1 Key concepts of the class

- Convex Optimization
- Global Optimization
- Machine Learning and function approximation

#### M.13.2.2 What is the purpose of this course?

Computational Intelligence serves as a combined multidisciplinary subject, encompassing a wide range of topics. These include numeric optimization, especially convex optimization, which is the necessary and required topic for most of the modern engineering and scientific work. The course also covers global non-convex optimization methods, which are important instruments in a number of areas: product design and manufacturing, control, and others. Basic information from a number of other areas, such as machine learning, are added to complete the picture of modern intelligent computational tools that serve the same set of goals.

#### M.13.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to outline:

- Convexity, Convex Sets, Convex optimization

- Quadratic programming, Second Order Cone Programming, Semidefinite Programming
- Optimization in Controller design, Optimization in path planning, Optimization in Mechanical Engineering
- Nonlinear non-convex optimization, RRT algorithm, Genetic Algorithm, Particle Swarm Optimization, Function approximation.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to understand:

- Structure of optimization problems
- Convexity criteria
- Properties of convex optimization
- Types of problems that cab be transformed into Linear Programs
- Types of problems that cab be transformed into Quadratic Programs
- Types of problems that cab be transformed into Second Order Cone Programs
- Types of problems that cab be transformed into Semidefinite Programs
- Types of non-convex problems that can be approximated by a convex relaxation
- Mixed-integer Optimization
- Path planning as an optimization
- Controller design as an optimization
- Optimal parameter choice
- RRT implementation
- PSO implementation

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Design Control using Convex Optimization.
- Implement iterative non-linear controllers with inequality constraints.
- Find optimal parameter choice for processes
- proof convexity of a problem
- program optimization in CVX

- make RRT-based algorithms, understand their limitations
- make PSO-based algorithms, understand their limitations
- make GA-based algorithms, understand their limitations

#### M.13.2.4 Course evaluation

**Table M.33:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 30 |
| Interim performance assessment | 30 | 20 |
| Exams | 50 | 50 |

#### M.13.2.5 Grades range

**Table M.34:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 85-100 |
| B. Good | 75-89 | 70-84 |
| C. Satisfactory | 60-74 | 50-69 |
| D. Poor | 0-59 | 0-49 |

#### M.13.2.6 Resources and reference material

Main textbooks:

- Engelbrecht, A.P., 2007. Computational intelligence: an introduction. John Wiley & Sons.
- Pedrycz, W., 1997. Computational intelligence: an introduction. CRC press.
- Konar, A., 2006. Computational intelligence: principles, techniques and applications. Springer Science & Business Media.

### M.13.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

Table M.35: Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction to optimization methods | 6 |
| 2 | Convex Optimization | 6 |
| 3 | Global Optimization | 6 |

#### M.13.3.1 Section 1

**Section title:** Introduction to optimization methods

**Topics covered in this section:**

- Optimization problem types.
- Constraint types.
- Cost function (Reward function) types.
- Practical examples of optimization problems.
- Basic optimization algorithms and their limitations.
- Lagrange multipliers.
- Gradient descent.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Describe an engineering problem as an optimization problem.
2. What is the domain of an optimization problem?
3. What is the difference between Convex and Non-convex optimization?

**Typical questions for seminar classes (labs) within this section**

1. Formulate a linear system with multi-dimensional solution space and inequality constraints.
2. Solve a linear system with inequality constraints via gradient descent.

### M.13.3.2 Section 2

**Section title:** Convex Optimization.

**Topics covered in this section:**

- Convex sets.
- Convex functions.
- Convex optimization problems.
- Properties of the convex optimizations.
- Linear Programming
- Quadratic Programming.
- Second Order Cone Programming.
- Semidefinite Programming.
- Vertical Stability of a bipedal robot as an optimization.
- Quadrotor path planning as an optimization.
- Controller design as an optimization.
- CVX.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Provide examples of convex problems.
2. What is a convex domain?
3. Examples of problems with a convex cost but non-convex domain?
4. What is the difference between quadratic and conic programming?
5. What is the hierarchy of convex optimization problems?
6. What solvers can solve quadratic programs?
7. What solvers can solve SOCP?
8. What solvers can solve LP?

9. What solvers can solve SDP?

**Typical questions for seminar classes (labs) within this section**

1. What kinds of inequality constraints make the problem non-convex?
2. What kinds of inequality constraints make the problem non-feasible?
3. Show an example of a problem where the cost is a 2-norm. Show that it is a SOCP.
4. Show an example of a problem where the cost is a 2-norm. Prove that it can be equivalently solved as a QP.
5. Solve trajectory planning for a quadrotor as a SOCP.
6. Solve vertical stability check for a biped as a QP.
7. implement ZMP trajectory planning as a QP.
8. implement LTI controller design as a SDP.

#### M.13.3.3 Section 3

**Section title:** Global Optimization

**Topics covered in this section:**

- Properties of non-convex problems.
- Problems with multiple solutions.
- Problems with weak local minima.
- Problems with computationally expensive gradients.
- Path planning on non-convex maps.
- Controller design as a non-convex problem.
- Relaxations.
- Mixed integer programming.
- PSO.
- RRT.
- Genetic Algorithm.
- Machine Learning.

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Provide examples of non-convex problems?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

2. Why non-convex problems can have multiple solutions?
3. What are local minima?
4. What is global optimization?
5. Provide an example of a non-convex path planning problems?
6. What are the limitations of PSO?

**Typical questions for seminar classes (labs) within this section**

1. Implement PSO for a parameter optimization problem.
2. Make a comparative study of PSO, GA and Random Search.
3. What is the difference between random search, Sobol sequences-based methods and PSO? Show it in the numerical examples.
4. Implement RRT
5. Implement GA

**Test questions for final assessment in this section**

1. Solve minimum distance to a plane problem, when the domain is non-convex.
2. Find optimal controller parameters for a given trajectory of a non-linear system.
3. Which non-linear algorithms can solve problems with non-convex domains?

## M.14 Computer Vision

- **Course name:** Computer Vision
- **Course number:** R-03

### M.14.1 Course Characteristics

#### M.14.1.1 Key concepts of the class

- Computer vision techniques
- Classical and deep learning models

#### M.14.1.2 What is the purpose of this course?

This course provides an intensive treatment of a cross-section of the key elements of computer vision, with an emphasis on implementing them in modern programming environments, and using them to solve real-world problems. The course will begin with the fundamentals of image processing and image filtering, but will quickly build to cover more advanced topics, including image segmentation, object detection and recognition, face detection, content-based image retrieval, artificial neural networks, convolutional neural networks, generative adversarial networks and much more. A key focus of the course is on providing students with not only theory but also hands-on practice of building their computer vision applications.

#### M.14.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to process the video

- Robots visual perception strategies
- Significant exposure to real-world implementations
- To develop research interest in the theory and application of computer vision

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to choose the correct computer vision model.

- Suitability of different computer vision models in different scenarios
- Ability to choose the right model for the given task

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to deploy and developed models.

- Hands on experience to implement different models to know inside behavior
- Sufficient exposure to train and deploy model for the given task
- Fine tune the deployed model in the real-world settings

#### M.14.1.4 Course evaluation

**Table M.36:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 20 |
| Interim performance assessment | 30 | 50 |
| Exams | 50 | 30 |

#### M.14.1.5 Grades range

**Table M.37:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 90-100 |
| B. Good | 75-89 | 75-89 |
| C. Satisfactory | 60-74 | 60-74 |
| D. Poor | 0-59 | 0-59 |

#### M.14.1.6 Resources and reference material

- Handouts supplied by the instructor
- Materials from the interment and research papers shared by instructor
-
-

## M.14.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.38:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Image Acquisition and Basic Image Processing | 8 |
| 2 | Image Filtering and Binary Vision | 8 |
| 3 | Feature Extractors and Descriptors | 16 |
| 4 | Deep Learning models for computer vision | 16 |

#### M.14.2.1 Section 1

**Section title:** Image Acquisition and Basic Image Processing

**Topics covered in this section:**

- Computer vision in action
- The Human Vision System
- Optical Illusions
- Sampling and Quantization
- Image Representation
- Colour Spaces

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. What are the color spaces and where it's used?
2. What are the primary and secondary colors?
3. How image is formed into computers?
4. How you will convert the RGB to grayscale images

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for seminar classes (labs) within this section**

1. Loading and plotting the images in python environment
2. Convertion of different color spaces
3. How you find the skin in the images based on the color space models
4. how to find red eye dot in face using color space models

**Test questions for final assessment in this section**

1. How you can distinguish different color spaces?
2. Explain and provide the reason for the blind spot creation in human eye.
3. In what scenarios computer vision is better than human vision?
4. Write down different robotic application areas where computer vision is applied successfully.

### M.14.2.2 Section 2

**Section title:** Image Filtering and Binary Vision

**Topics covered in this section:**

- Image noise
- Convolutions and kernels
- Smoothing and blurring
- Thresholding and histograms
- Morphological operations
- Gradients and Edge detection

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. What are the challenges to perform histogram task?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

2. Apply convolutional filter to calculate the response.
3. What kind of parameters are required to apply different image filters?
4. How you will compute the gradients of the image and its benefits?

**Typical questions for seminar classes (labs) within this section**

1. Implement Otsu Method
2. Implement Sobel, Preweitt filters
3. Implement Canny edge detector
4. Perform analysis over the different filtering on the given images

**Test questions for final assessment in this section**

1. Calculate the kernels for the given images
2. Explain the difference between different filters
3. What is image noise and how it contributes to make the computer vision task difficult?
4. Apply different combination of the filters to achieve the required output of the given image.

#### M.14.2.3 Section 3

**Section title:** Feature Extractors and Descriptors

**Topics covered in this section:**

- Histogram of Gradients (HoG)
- Scale-invariant feature transform (SIFT)
- Harris corner detector
- Template matching

- Bag of visual words
- Face Detection and Recognition (Viola Johns)

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How feature extractor works over the given image?
2. What is the difference between the feature extraction and descriptors?
3. Explain the examples of descriptors and feature extractors.
4. Write down the pros and cons of SIFT, HOG and Harris.

**Typical questions for seminar classes (labs) within this section**

1. Implement template matching algorithm
2. Implement histogram of gradient using CV2 library
3. Implement of SIFT for the given task
4. Implement Harris corner detection
5. Analysis of different extractors for the given task

**Test questions for final assessment in this section**

1. How you distinguish different feature extractors and descriptors?
2. What are the possible methods to detect the corners?
3. How corners are useful to help the robotic vision task?
4. How you will patch the different images to construct the map of the location?

### M.14.2.4 Section 4

**Section title:** Deep learning models for computer vision

**Topics covered in this section:**

- You Only Look Once: Unified, Real-Time Object Detection (YOLO)
- Generative Adversarial Networks (GAN)
- Fully Convolutional Networks (FCN) for semantic segmentation
- Multi Domain Network (MDNet) for object tracking
- Generic Object Tracking Using Regression Networks (GOTURN) for object tracking

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How classification task is different from detection task?
2. Explain the transfer learning mechanism for object detection task.
3. How many types of model exist for object tracking in videos.
4. Write down the pros and cons of YOLO, FCN and MDNet.

**Typical questions for seminar classes (labs) within this section**

1. Implement YOLO using transfer learning mechanism
2. Implement GAN for MNIST dataset
3. Implement FCN and GOTURN
4. Analysis of different models for the given task

**Test questions for final assessment in this section**

1. What are the loss functions used in YOLO?
2. What are the learnable parameters of FCN for semantic segmentation?
3. How semantic segmentation is different from instance segmentation?
4. Write the application areas for object tracking in robotics.

## M.15 Cybercrime and Forensics

- Course name: Cybercrime and Forensics
- Course number: SNE-???

### M.15.1 Course characteristics

### M.15.2 Key concepts of the class

- Law, regulations and modern tendencies of the high-tech crimes
- Computer forensics approaches and techniques
- Incident response and threat hunting methods

### M.15.3 What is the purpose of this course?

Modern tactics and techniques of high-tech crimes, including counter -forensics methods, are evolving rapidly according to the past several years. Therefore, the purpose of this course is to provide for students the necessary knowledge and abilities to obtain and analyze digital evidence in a way to provide investigations that will comply with the current law and regulations. Another purpose for the course is to learn for students how to counteract with ongoing computer incidents, intrusions and to perform threat hunting in the computer systems

### M.15.4 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to identify and define

- Methods for investigating and responding to cybersecurity incidents
- Main types of computer attacks and the technical and non-technical techniques used by attackers;
- Compliance requirements to produce valid computer-technical expertise for further legal procedures;

- Aquisition techniques depending on the affected digital media and environment conditions
- Computer systems' artifacts that were affected during the incident
- Specific hardware and software forensic tools depending on the type of incident;
- Decryption and decoding methods for protected and hidden data, methods of counter-forensics technology.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- Difference between different types of computer incidents
- The difference in compliance requirements for specific cybercrime cases
- Computer attacker model and kill chain tactics
- Filesystems analysis methods
- Volatile memory analysis methods
- Network analysis methods
- Malware analysis methods

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to demonstrate

- Organizing an incident response to a cybersecurity incident and minimize potential damage
- Determination of the type and causes of the incident
- Determination of the computer systems' artifacts that are required for the acquisition
- Collection of digital evidence and proper documentation of it
- Recovered deleted and hidden information
- Restored an incident chronology during the investigation, determination of the methods used by the attacker and the impact on the attacked system
- Conduction of investigation on various types of computer attacks
- Conduction malware analysis
- Correct and efficient usage of open source forensics software and hardware

### M.15.5 Course evaluation

**Table M.39:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 20 |
| Project | 30 | 60 |
| Exam | 50 | 20 |

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

### M.15.6 Grades range

**Table M.40:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 90-100 |
| B. Good | 75-89 | 70-89 |
| C. Satisfactory | 60-74 | 60-69 |
| D. Poor | 0-59 | 0-59 |

If necessary, please indicate freely your course's grading features:

The laboratory assignments are mandatory with a required minimum result of 6/10 for each - including re-takes and late submissions - to complete the course. The semester starts with the default range as proposed in the Table above, but it may changes slightly depending on how the semester progresses.

### M.15.7 Resources and reference material

- "Practical forensic imaging. Securing digital evidence with Linux tools". Bruce Nikkel
- "Incident response and computer forensics". K.Mandia, C.Prosise, and M.Pepe
- "Digital Evidence and Computer Crime". Eoghan Casey

**Table M.41:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Modern high-tech crimes and the law | 4 |
| 2 | Data acquisition and securing digital evidence | 4 |
| 3 | Computer systems' artifacts and their analysis methods | 6 |
| 4 | Volatile data analysis methods | 6 |
| 5 | Incident response and threat hunting | 4 |
| 6 | Labs | 56 |

### M.15.8 Course Sections

### M.15.9 Section 1

**Section title:** Modern high-tech crimes and the law

**Topics covered in this section:**

- Law, regulations and modern tendencies of the high-tech crimes
- Computer forensics approaches and techniques
- Incident response and threat hunting methods

**What forms of the evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

- What are the typical attacks which can be used against the banking system?
- What is the attacker model?
- What is the computer incident?

- What types of incidents can lead to criminal code articles for an attacker?

**Typical questions for seminar classes (labs) within this section**

- Identify risks and develop mitigation techniques before acquiring evidence for a given case
- Develop an attacker model for a specific incident
- Identify the most important compliance requirements for preservation evidence in the court case?

**Test questions for final assessment in this section**

- As above

## M.15.10 Section 2

**Section title:** Data acquisition and securing digital evidence

**Topics covered in this section:**

- Compliance requirements for the evidence acquisition
- Non-volatile data evidence collection
- Volatile data evidence collection
- Securing digital evidence with open source tools

**What forms of the evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

- What are the pros and cons of using software or hardware tools for acquisition?
- What are the important steps to perform data evidence acquisition on the live system?
- What is the difference between non-volatile and volatile data from the perspective of computer forensics?
- What are the legal aspects of preparing before conducting computer forensic analysis based on the positions and responsibilities of forensic investigators?
- What kind of computer systems' components would be less important during a live acquisition?

**Typical questions for seminar classes (labs) within this section**

- Depending on the incident define software and hardware that can be used to collect and preserve digital evidence
- Collect the evidence on a virtual environment
- Collect the evidence from the live system
- Collect the evidence of the volatile data
- Provide integrity, confidentiality, and non-repudiation for acquired evidence

**Test questions for final assessment in this section**
As above

### M.15.11 Section 3

**Section title:**

**Topics covered in this section:** Computer systems' artifacts and their analysis methods

- Anti-forensics methods and recovery information
- Windows forensics
- Filesystem forensics

**What forms of the evaluation were used to test students' performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

- What kind of methods do you know for an attacker to hide and delete information?
- What are the important artifacts that can be used for the analysis of the Windows systems?
- What is the difference between DEFT and CAIN software forensics distributions?
- What is MAC time?
- What is the conceptual difference between FAT and NTFS?

**Typical questions for seminar classes (labs) within this section**

- Analyze the incident that involves the USB stick of the attacker
- Create a timeline based on the timestamps of the artifacts
- Find and recover hidden information on the hard drive
- Extract and analyze filesystem journals
- Find encrypted information
- Identify the slack spaces that contain deleted data

**Test questions for final assessment in this section**

As above

### M.15.12 Section 4

**Section title:** Volatile data analysis methods

**Topics covered in this section:**

- Operating memory forensics

- Network forensics

**What forms of the evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

- What is fileless malware?
- How can rootkits affect the evidence?
- What kind of operating memory artifacts can be useful for cybercrime investigation?
- What is difficult about dumping a memory?
- What is difficult about dumping network traffic?

**Typical questions for seminar classes (labs) within this section**

- Identify direct kernel object manipulation in the given sample
- Find unlinking from the active process list
- Trace and detect used cryptographical keys on the incident
- Determine the original source of an attacker's compromise on the given network traffic
- Establish and present a timeline of the attacker's activities for a specific case

**Test questions for final assessment in this section**

As above

## M.15.13 Section 5

**Section title:** Incident response and threat hunting

**Topics covered in this section:**

- Incident response techniques
- Sandboxing
- Malware analysis
- Threats hunting

**What forms of the evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

- What limitations might you have during the incident response?
- What type of incident responses can be provided during the incident
- What is the difference between incident response and computer forensics in general
- What is sandboxing and how it could be used in the incident response?
- What type of threats can occur for investigators during investigation?

**Typical questions for seminar classes (labs) within this section**

- Identify the methods that can detect anomaly behavior for a typical Windows system processes
- Identify persistence mechanisms that are used by the given malicious process

- Identify illegitimate network activity on the given network traffic
- Develop an effective sandboxing environment for malware detection and examination of its behavior
- Develop indicators of compromise to detect threats on multiple systems

**Test questions for final assessment in this section**

As above

## M.16 Distributed Systems

- **Course name:** Distributed Systems
- **Course number:** XYZ

### M.16.1 Course Characteristics

#### M.16.1.1 Key concepts of the class

- Principles of designing and implementing distributed systems
- Structure of distributed system, and algorithms used to implement distributed system

#### M.16.1.2 What is the purpose of this course?

This is an introductory course in distributed systems. During the course, students will learn the fundamental principles and techniques that can be applied to design and develop distributed systems, and will cover architectures of distributed systems, communication, naming, fault tolerance, consistency replication, virtualization and security. In addition, we will discuss different aspects of design and implementation of popular distributed systems (such as bittorrent, google file system, HDFS, etc.), programming models (MapReduce, MapReduce2/YARN) and consensus algorithms (Raft and Paxos). The course will not only cover computer science related topics, but will also include a substantial part of software engineering activities.

#### M.16.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to understand distributed systems

- Key principles involved in designing and implementing distributed systems
- Distributed Architectures and their management
- Canonical problems and solutions for distributed systems

- Resource sharing, replication and consistency algorithms in distributed environments
- Virtualization, Orchestration and cloud management

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to understand the key components of distributed systems

- Role of different components in distributed systems
- Role of virtualization in distributed systems
- Management and Orchestration of resources in distributed systems
- Fault tolerant approaches in distributed systems

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to develop and implement different components in a distributed environment

- Implementation of access and location transparency in distributed systems
- Designing and implementing consensus algorithms for different distributed environments
- Management of concurrent tasks in distributed settings
- Designing fault tolerant distributed systems

#### M.16.1.4 Course evaluation

**Table M.42:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 20 |
| Interim performance assessment | 30 | 25 |
| Exams | 50 | 55 |

**Table M.43:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 90-100 |
| B. Good | 75-89 | 75-89 |
| C. Satisfactory | 60-74 | 60-74 |
| D. Poor | 0-59 | 0-59 |

#### M.16.1.5 Grades range

#### M.16.1.6 Resources and reference material

- 

### M.16.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.44:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction to distributed systems | 16 |
| 2 | Virtualization and cloud computing | 16 |
| 3 | Canonical Problems and Solutions | 24 |

#### M.16.2.1 Section 1

**Section title:** Introduction to distributed systems

**Topics covered in this section:**

- Distributed architectures
- Types of distributed systems
- Processes & Threads
- Multiprocessor and distributed scheduling
- Communication in distributed systems
- Naming in distributed systems

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. State advantages of object based architectures over layer based architectures in distributed system?
2. State advantages of the random walk approach over the flooding approach in unstructured P2P networks for locating the data?
3. Why global variables are not allowed in a RPC?
4. Describe approaches for a server to handle incoming socket connections (at least two).
5. What are pros and cons of different approaches of sockets? When would you choose which one?

**Typical questions for seminar classes (labs) within this section**

1. Find the process ID (PID) of the running program and list of threads associated to that process?
2. Implementation of new threads
3. Performance analysis of given task
4. Implementation of Process and Global Interpreter Lock (GIL)

**Test questions for final assessment in this section**

1. Explain difference between threads and process
2. Define Global Interpreter Lock (GIL) and its role
3. What are remote method invocation (RMI). Please state the benefits and challenges of RMI.
4. Define the role of Marshaling and Unmarshaling in RPC
5. State at least two benefits of both Iterative and recursive naming resolution schemes.

#### M.16.2.2 Section 2

**Section title:** Virtualization and cloud computing

**Topics covered in this section:**

- Foundations of virtualization
- OS-level virtualization
- System level virtualization
- Memory virtualization
- Cloud and data centres

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are containers? State at least three benefits of using them?
2. Why memory reclamation approaches are needed for virtual machines?
3. What are "cgroup" and "namespace" subsystems. How can we use the "cgroup" subsystem?
4. Briefly explain about the "Ballooning" approach and its working to reclaim VM memory?
5. What is a Unikernel? State at least two benefits and one drawback.

**Typical questions for seminar classes (labs) within this section**

1. Hands on Amazon Elastic Compute Cloud (EC2)
2. Set up a public address in Elastic IPs tab and assign it to the web server on which WordPress is hosted.
3. Configure WebServer so that your blog page is accessible only through the public IP address (without specifying the /wordpress path). For example, you specify a public IP address in the address bar of the browser and gain access to the blog.

4. There may be problems with CSS. Search the web for how to fix broken CSS after changing the site URL.

**Test questions for final assessment in this section**

1. State two benefits of immersion cooling over traditional air cooling in data centers?
2. State one main difference of a container from a System Virtual machine?
3. Briefly explain about Copy-on-write storage.
4. What are the roles of “ISA” and “ABI” in operating systems?
5. What is the biggest drawback of Google File system over CephFS?

### M.16.2.3 Section 3

**Section title:** Canonical Problems and Solutions

**Topics covered in this section:**

- Mutual exclusion
- Leader election,
- Clock synchronization
- Consistency issues
- Caching and replication
- Fault Tolerance

**What forms of evaluation were used to test students’ performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Why is Anti-Entropy Protocol considered better than the Gossiping protocol for achieving consistency.

2. What is the role of Heartbeats in RAFT?
3. Explain PAXOS to achieve consistency?
4. What is the benefit of three phase commit over two phase commit? Please specify in terms of coordinator failure?
5. How many replicas are required to identify the fault in Byzantine failure scenarios?

**Typical questions for seminar classes (labs) within this section**

1. In the replication set of databases, what problem may appear if we have an even number of nodes?
2. ICRUD operations in MongoDB.
3. Create simple chat web-application which uses replica set.
4. Explain the steps to shutdown all VPS instances.

**Test questions for final assessment in this section**

1. What is a schema-less data model
2. Why do we need NoSQL? What are its benefits over SQL in terms of ACID and BASE properties?
3. Why is version control recommended? Name at least two version control systems?
4. Role of recovery line and check pointing in distributed snapshot?
5. Explain the difference between Lamport and vector clocks?
6. How the use of buffers at receivers enhance the QoS in networks for streaming media applications? Briefly explain using an example.

# M.17 Dynamics of Non Linear Robotic Systems

- **Course name:** Dynamics of Non Linear Robotic Systems
- **Course number:** R-01

## M.17.1 Course Characteristics

### M.17.1.1 Key concepts of the class

Robotics; Robotic components; Robotic control.

### M.17.1.2 What is the purpose of this course?

This course is an introduction to the field of robotics. It covers the fundamentals of kinematics, dynamics, and control of robot manipulators, robotic vision, and sensing. The course deals with forward and inverse kinematics of serial chain manipulators, the manipulator Jacobian, force relations, dynamics, and control. It presents elementary principles on proximity, tactile, and force sensing, vision sensors, camera calibration, stereo construction, and motion detection. The course concludes with current applications of robotics in active perception, medical robotics, autonomous vehicles, and other areas.

### M.17.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to ...

- Model the kinematics of robotic systems.
- Compute end-effector position and orientation from joint angles of a robotic system.
- Compute the joint angles of a robotic system to reach the desired end-effector position and orientation.
- Compute the linear and angular velocities of the end-effector of a robotic system from the joint angle velocities.

- Convert a robot's workspace to its configuration space and represent obstacles in the configuration space.
- Compute valid path in a configuration space with motion planning algorithms.
- Apply the generated motion path to the robotic system to generate a proper motion trajectory.
- Apply the learned knowledge to several robotic systems: including robotic manipulators, humanoid robots.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to ...

- Name various applications of robots
- Describe the current and potential economic and societal impacts of robot technology
- Use the Jacobian to transform velocities and forces from joint space to operational space
- Determine the singularities of a robot manipulator
- Formulate the dynamic equations of a robot manipulator in joint space and in Cartesian space
- List the major design parameters for robot manipulators and mobile robots
- List the typical sensing and actuation methods used in robots
- Analyze the workspace of a robot manipulator
- List the special requirements of haptic devices and medical robots
- Effectively communicate research results

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

- Describe rigid body motions using positions, orientations, frames, and mappings
- Describe orientations using Euler angles, fixed angles, and quaternions
- Develop the forward kinematic equations for an articulated manipulator

- Describe the position and orientations of a robot in terms of joint space, Cartesian space, and operational space
- Develop the Jacobian for a specific manipulator
- Determine the singularities of a robot manipulator
- Write the dynamic equations of a robot manipulator using the Lagrangian Formulation
- Analyze the workspace of a robot manipulator

#### M.17.1.4 Course evaluation

**Table M.45:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Weekly quizzes | 20 | |
| Home assignments | 20 | |
| Project | 20 | |
| Midterm Exam | 20 | |
| Final Exam | 20 | |

#### M.17.1.5 Grades range

**Table M.46:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 92-100 | |
| B. Good | 80-91 | |
| C. Satisfactory | 65-79 | |
| D. Poor/Fail | 0-59 | |

#### M.17.1.6 Resources and reference material

- Siciliano, Sciavicco, Villani, and Oriolo, Robotics: Modeling, Planning and Control, Springe

## M.17.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.47:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction to robotics | 14 |
| 2 | Kinematics | 14 |
| 3 | Differential kinematics | 16 |
| 4 | Dynamics | 16 |

### M.17.2.1 Section 1

**Section title:** Introduction to robotics

**Topics covered in this section:**

- Introduction to Robotics, History of Robotics
- Introduction to Drones
- Introduction to Self driving cars
- Programming of Industrial Robot

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the difference between the manipulator arm and manipulator wrist
2. What is Node in ROS
3. What are the disadvantages of ROS
4. Write sensors which are used in self driving cars.
5. Describe the classical approach for deign self driving car

### Typical questions for seminar classes (labs) within this section

1. Advantages and drawbacks of robotic manipulators
2. Programming industrial robots
3. Developing self driving car
4. Drones and controllers for them

### Test questions for final assessment in this section

1. Typical commands for programming industrial manipulator motions
2. Types of robots and their application ares
3. Control of self driving car

## M.17.2.2 Section 2

**Section title:** Kinematics

### Topics covered in this section:

- Rigid body and Homogeneous transformation
- Direct Kinematics
- Inverse Kinematics

### What forms of evaluation were used to test students' performance in this section?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

### Typical questions for ongoing performance evaluation within this section

1. Properties of Rotation Matrix
2. How to find Euler angles from rotation matrix
3. How to compute rotation matrix from knowing Euler angles
4. How to derive equations for direct kinematic problem
5. How to solve inverse kinematics problem

**Typical questions for seminar classes (labs) within this section**

1. Structure, properties, and advantages of Homogeneous transformation
2. Expression for rotation around an arbitrary axis
3. Euler angles
4. Difference between Joint and Operational spaces
5. Direct kinematics for serial kinematic chain
6. Piper approach for inverse kinematics

**Test questions for final assessment in this section**

1. Transformation between reference frames
2. Find Euler angles for given orientation matrix and transformation order
3. Transformation between Cartesian and operational spaces
4. Direct kinematic for SCARA robot
5. Inverse kinematic for SCARA robot

### M.17.2.3 Section 3

**Section title:** Differential kinematics

**Topics covered in this section:**

- Differential kinematics
- Geometric calibration
- Trajectory Planning

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Write the matrix of differential transformation

2. What is Jacobian matrix
3. Difference between parametric and non-parametric robot calibration.
4. Why we need complete and irreducible model
5. How trajectory planning is realised
6. What is trajectory junction

**Typical questions for seminar classes (labs) within this section**

1. Jacobian matrix calculation
2. Jacobian matrices for typical serial manipulators
3. Robot calibration procedure
4. complete, irreducible geometric model
5. robot control strategies with offline errors compensation
6. Trajectory planning in joint and Cartesian spaces
7. Trajectory junction

**Test questions for final assessment in this section**

1. Write Jacobian for Polarrobot
2. Advantages and disadvantages parametric and non-parametric robot calibration.
3. complete, irreducible geometric model for spherical manipulator
4. Compute the joint trajectory q(t) from q(0) = 1 to q(2) = 4 with null initial and final velocities and accelerations. (polynomial)
5. Obtain manipulator trajectory for given manipulator kinematics, initial and final states and velocity and acceleration limits/

#### M.17.2.4 Section 4

**Section title:** Dynamics

**Topics covered in this section:**

- Dynamics of Rigid body
- Lagrange approach
- Newton-Euler approach

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Energy of rigid body
2. Dynamics of rigid body
3. What is Direct and Inverse Dynamics
4. Difference between Newton Euler and Lagrange Euler approaches

**Typical questions for seminar classes (labs) within this section**

1. Dynamics of rigid body
2. Direct and Inverse Dynamic
3. Newton-Euler Approach
4. Lagrange-Euler Approach

**Test questions for final assessment in this section**

1. Solve inverse dynamics problem for Cartesian robot
2. Solve direct dynamics problem for RRR spherical manipulator
3. Moving frame approach for dynamics modelling

## M.18 High-Dimensional Data Analysis

- **Course name:** High-Dimensional Data Analysis
- **Course number:** DS-06
- **Area of instruction:** Computer Science and Engineering

### M.18.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 1st year of MSc
- **Semester of instruction:** 1st semester
- **No. of Credits:** 5 ECTS
- **Total workload on average:** 180 hours overall
- **Frontal lecture hours:** 2 hours per week.
- **Frontal tutorial hours:** 0 hours per week.
- **Lab hours:** 2 hours per week.
- **Individual lab hours:** 2 hours per week.
- **Frequency:** weekly throughout the semester.
- **Grading mode:** letters: A, B, C, D.

### M.18.2 Course outline

This course gives the knowledge in data analysis and interpretation. It starts by learning the mathematical definition of distance and use this to motivate the use of the singular value decomposition (SVD) for dimension reduction and multi-dimensional scaling and its connection to principle component analysis. It also describes the principal component analysis and factor analysis and demonstrates how these concepts are applied to data visualization and data analysis of high-throughput experimental data. Moreover, the course gives a brief introduction to machine learning and apply it to high-throughput data. It presents the general idea behind clustering analysis and descript K-means and hierarchical clustering and demonstrate how these are used in describe prediction algorithms such as k-nearest neighbors along with the concepts of training sets, test sets, error rates and cross-validation. The students will be required to participate in laboratory practicum and solve practical tasks using hardware and Python environment.

## M.18.3 Expected learning outcomes

- Apply different data analysis for dimension reduction and multi-dimensional scaling
- Be able to select best data analysis approach for a particular problem
- Be familiar with principal component analysis and factor analysis and understand how these concepts are applied to data visualization and data analysis of high-throughput experimental data

## M.18.4 Required background knowledge

Strong mathematical background in Calculus, Linear Algebra, Differential Equations, Statistics and Numerical Methods as well as programming in Python and C/C++.

## M.18.5 Prerequisite courses

- Calculus I
- Calculus II
- Linear Algebra
- Differential Equations
- Numerical Methods
- Statistics

## M.18.6 Detailed topics covered in the course

- Mathematical Distance
- Dimension Reduction
- Singular Value Decomposition and Principal Component Analysis
- Multiple Dimensional Scaling Plots
- Factor Analysis
- Dealing with Batch Effects

- Clustering
- Heatmaps
- Basic Machine Learning Concepts

### M.18.7 Textbook

- T. Tony Cai, Xiaotong Shen, ed. (2011). High-dimensional data analysis. Frontiers of Statistics. Singapore: World Scientific
- Christophe Giraud (2015). Introduction to High-Dimensional Statistics. Philadelphia: Chapman and Hall/CRC

### M.18.8 Reference material

- Peter Bühlmann and Sara van de Geer (2011). Statistics for high-dimensional data: methods, theory and applications. Heidelberg; New York: Springer
- Slides will be provided during the course

### M.18.9 Required computer resources

NA

### M.18.10 Evaluation

- Final Project (40%)
- Assignments (40%)
- Midterm Exam (20%)

# M.19 Large Systems

- **Course name:** Large Systems
- **Course number:** xyz

## M.19.1 Course Characteristics

### M.19.1.1 Key concepts of the class

- Virtualization and management of large systems
- Communication, coordination and fault tolerance IT for infrastructure

### M.19.1.2 What is the purpose of this course?

In large organizations complex architectures are built where products and protocols of multiple vendors have to inter-operate. The course focuses on understanding the challenges and building a scalable IT infrastructure that is flexible and efficient to manage. Topics covered are: Virtualization and Cloud Computing, Workstation and Server deployment, Data-centers and Infrastructure management methodologies.

### M.19.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember the followings:

- Install and configure large systems
- Automate the installation and the deployment process in large systems
- Usage of virtualized systems.
- Manage scalable infrastructures
- Fault tolerance and solutions in large systems

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to understand the key components of large systems

- Management and control of large systems
- Different components and architectures of large systems
- Role of virtualization and orchestration for managing large systems.
- Continuous integration and continuous deployment approaches

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to develop, manage and implement different large system components

- Implementation of different virtualization techniques over different platforms
- Designing and implementing orchestration schemes for large systems
- Implement Continuous integration and continuous deployment to automate the installation and the deployment.
- Designing fault tolerant systems

#### M.19.1.4 Course evaluation

**Table M.48:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 35 |
| Interim performance assessment | 30 | 35 |
| Exams | 50 | 30 |

#### M.19.1.5 Grades range

**Table M.49:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 90-100 |
| B. Good | 75-89 | 75-89 |
| C. Satisfactory | 60-74 | 60-74 |
| D. Poor | 0-59 | 0-59 |

#### M.19.1.6 Resources and reference material

- 
- 
- 

### M.19.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.50:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Virtualization | 16 |
| 2 | Building blocks for Large systems | 16 |
| 3 | Automation and Management | 16 |

#### M.19.2.1 Section 1

**Section title:** Virtualization

**Topics covered in this section:**

- Foundations of virtualization
- OS-level virtualization
- System level virtualization
- Memory virtualization
- Orchestration and Management

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are containers? State at least three benefits of using them?
2. Why memory reclamation approaches are needed for virtual machines?
3. What are “cgroup” and “namespace” subsystems. How can we use the “cgroup” subsystem?
4. Briefly explain about the “Ballooning” approach and its working to reclaim VM memory?
5. What is a Unikernel? State at least two benefits and one drawback.

**Typical questions for seminar classes (labs) within this section**

1. Validate that when a node goes down a new instance is launched. Show how the redistribution of the instances can happen when the broken node comes back alive?
2. Deploy a simple application (e.g. a simple web page showing the hostname of the host node it is running upon) for each orchestrator and validate that its instances spreads around the container farm.
3. What is the difference between simple load-balancing and HA load-balancing?
4. Install at least 4 different types of guest systems or distributions on the physical host?
5. Create a simple web page (preferably containing some pictures), construct an image with operating system you chose and deploy it via the unikernal.

**Test questions for final assessment in this section**

1. State two benefits of immersion cooling over traditional air cooling in data centers?
2. State one main difference of a container from a System Virtual machine?
3. Briefly explain about Copy-on-write storage.
4. What are the roles of “ISA” and “ABI” in operating systems?
5. What is the biggest drawback of Google File system over CephFS?

### M.19.2.2 Section 2

**Section title:** Building blocks for Large systems

**Topics covered in this section:**

- Storage technologies
- Backup and monitoring technologies
- NoSQL and distributed storage
- Communication and coordination approaches in LS
- Fault tolerance in LS

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is a schema-less data model?
2. Why do we need NoSQL? What are its benefits over SQL in terms of ACID and BASE properties?
3. Why is Anti-Entropy Protocol considered better than the Gossiping protocol for achieving consistency.
4. What is the role of Heartbeats in RAFT?
5. Explain PAXOS to achieve consistency?
6. How many replicas are required to identify the fault in Byzantine failure scenarios?

**Typical questions for seminar classes (labs) within this section**

1. Create the testing environment to test VM monitoring through a running web page.
2. Deploy zabbix server+agent (server is better to deploy on dedicated VM instance).

3. Use load-testing tool (yandex.tank, siege, k8, etc) and perform 2-3 different load tests. Report by varying the number of threads, number of clients, request body while performing requests.
4. Why do you think it is better to choose networked block device than networked file-systems, in terms of performance?

**Test questions for final assessment in this section**

1. same as above.

### M.19.2.3 Section 3

**Section title:** Automation and Management in LS

**Topics covered in this section:**

- IT management approaches
- DevOps and Agile practices
- Continuous Integration (CI) for automation
- Continuous Deployment (CD) for automation
- Data centres- Hardware technologies
- Blockchain technology

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are continuous integration and continuous deployment?
2. Why is version control recommended? Name at least two version control systems?
3. State four differences between the Scrum and Kanban models?

4. Briefly explain about the Agile software development approach. How is it different from the waterfall model?

**Typical questions for seminar classes (labs) within this section**

1. Validate cold migration of your four guests.
2. Validate hot migration of your four guests.
3. What kind of general stages are there in the CI/CD pipeline? list them in the correct sequence and then describe them.
4. Select a CI/CD solution/platform and describe the project that you will apply CI/CD to it.
5. Try to automate the deployment of your project if the CI/CD pipeline passes.

**Test questions for final assessment in this section**

1. same as above.

## M.20 Machine Learning

- **Course name:** Machine Learning
- **Course number:** R-01

### M.20.1 Course characteristics

#### M.20.1.1 Key concepts of the class

- Machine learning paradigms
- Machine Learning approaches, and algorithms

#### M.20.1.2 What is the purpose of this course?

There is a growing business need of individuals skilled in artificial intelligence, data analytics, and machine learning. Therefore, the purpose of this course is to provide students with an intensive treatment of a cross-section of the key elements of machine learning, with an emphasis on implementing them in modern programming environments, and using them to solve real-world data science problems.

#### M.20.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?** By the end of the course, the students should be able to recognize and define

- Different learning paradigms
- A wide variety of learning approaches and algorithms
- Various learning settings
- Performance metrics
- Popular machine learning software tools

**- What should a student be able to understand at the end of the course?** By the end of the course, the students should be able to describe and explain (with examples)

- Difference between different learning paradigms
- Difference between classification and regression

- Concept of learning theory (bias/variance tradeoffs and large margins etc.)
- Kernel methods
- Regularization
- Ensemble Learning
- Neural or Deep Learning

**- What should a student be able to apply at the end of the course?** By the end of the course, the students should be able to apply

- Classification approaches to solve supervised learning problems
- Clustering approaches to solve unsupervised learning problems
- Ensemble learning to improve a model's performance
- Regularization to improve a model's generalization
- Deep learning algorithms to solve real-world problems

#### M.20.1.4 Course evaluation

**Table M.51:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 0 |
| Interim performance assessment | 30 | 40 |
| Exams | 50 | 60 |

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

#### M.20.1.5 Grades range

**Table M.52:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table **??**, but it may change slightly (usually reduced) depending on how the semester progresses.

#### M.20.1.6 Resources and reference material

- T. Hastie, R. Tibshirani, D. Witten and G. James. *An Introduction to Statistical Learning. Springer 2013.*
- *T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning. Springer 2011.*
- *Tom M Mitchel. Machine Learning, McGraw Hill*
- *Christopher M. Bishop. Pattern Recognition and Machine Learning, Springer*

### M.20.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.53:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Supervised Learning | 24 |
| 2 | Decision Trees and Ensemble Learning | 8 |
| 3 | Unsupervised Learning | 8 |
| 4 | Deep Learning | 12 |

#### M.20.2.1 Section 1

**Section title:** Supervised Learning

**Topics covered in this section:**

- Introduction to Machine Learning
- Derivatives and Cost Function
- Data Pre-processing
- Linear Regression
- Multiple Linear Regression
- Gradient Descent

- Polynomial Regression
- Splines
- Bias-varaince Tradeoff
- Difference between classification and regression
- Logistic Regression
- Naive Bayes
- Bayesian Network
- KNN
- Confusion Metrics
- Performance Metrics
- Regularization
- Hyperplane Based Classification
- Perceptron Learning Algorithm
- Max-Margin Classification
- Support Vector Machines
- Slack Variables
- Lagrangian Support Vector Machines
- Kernel Trick

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**What forms of evaluation were used to test students' performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

1. Is it true that in simple linear regression $R^2$ and the squared correlation between X and Y are identical?
2. What are the two assumptions that the Linear regression model makes about the **Error Terms**?
3. Fit a regression model to a given data problem, and support your choice of the model.

4. In a list of given tasks, choose which are regression and which are classification tasks.
5. In a given graphical model of binary random variables, how many parameters are needed to define the Conditional Probability Distributions for this Bayes Net?
6. Write the mathematical form of the minimization objective of Rosenblatt's perceptron learning algorithm for a two-dimensional case.
7. What is perceptron learning algorithm?
8. Write the mathematical form of its minimization objective for a two-dimensional case.
9. What is a max-margin classifier?
10. Explain the role of slack variable in SVM.

**Typical questions for seminar classes (labs) within this section**

1. How to implement various regression models to solve different regression problems?
2. Describe the difference between different types of regression models, their pros and cons, etc.
3. Implement various classification models to solve different classification problems.
4. Describe the difference between Logistic regression and naive bayes.
5. Implement perceptron learning algorithm, SVMs, and its variants to solve different classification problems.
6. Solve a given optimization problem using the Lagrange multiplier method.

**Test questions for final assessment in this section**

1. What does it mean for the standard least squares coefficient estimates of linear regression to be *scale equivariant*?
2. Given a fitted regression model to a dataset, interpret its coefficients.
3. Explain which regression model would be a better fit to model the relationship between response and predictor in a given data.
4. If the number of training examples goes to infinity, how will it affect the bias and variance of a classification model?

5. Given a two dimensional classification problem, determine if by using Logistic regression and regularization, a linear boundary can be estimated or not.
6. Explain which classification model would be a better fit to for a given classification problem.
7. Consider the Leave-one-out-CV error of standard two-class SVM. Argue that under a given value of slack variable, a given mathematical statement is either correct or incorrect.
8. How does the choice of slack variable affect the bias-variance tradeoff in SVM?
9. Explain which Kernel would be a better fit to be used in SVM for a given data.

#### M.20.2.2 Section 2

**Section title:** Decision Trees and Ensemble Methods

**Topics covered in this section:**

- Decision Trees
- Bagging
- Boosting
- Random Forest
- Adaboost

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**What forms of evaluation were used to test students' performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

1. What are pros and cons of decision trees over other classification models?
2. Explain how tree-pruning works.
3. What is the purpose of ensemble learning?
4. What is a bootstrap, and what is its role in Ensemble learning?
5. Explain the role of slack variable in SVM.

**Typical questions for seminar classes (labs) within this section**

1. Implement different variants of decision trees to solve different classification problems.
2. Solve a given classification problem problem using an ensemble classifier.
3. Implement Adaboost for a given problem.

**Test questions for final assessment in this section**

1. When a decision tree is grown to full depth, how does it affect tree's bias and variance, and its response to noisy data?
2. Argue if an ensemble model would be a better choice for a given classification problem or not.
3. Given a particular iteration of boosting and other important information, calculate the weights of the Adaboost classifier.

### M.20.2.3 Section 3

**Section title:** Unsupervised Learning

**Topics covered in this section:**

- K-means Clustering
- K-means++
- Hierarchical Clustering
- DBSCAN
- Mean-shift

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**What forms of evaluation were used to test students' performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

1. Which implicit or explicit objective function does K-means implement?
2. Explain the difference between k-means and k-means++.
3. Whaat is single-linkage and what are its pros and cons?
4. Explain how DBSCAN works.

**Typical questions for seminar classes (labs) within this section**

1. Implement different clustering algorithms to solve to solve different clustering problems.
2. Implement Mean-shift for video tracking

**Test questions for final assessment in this section**

1. K-Means does not explicitly use a fitness function. What are the characteristics of the solutions that K-Means finds? Which fitness function does it implicitly minimize?
2. Suppose we clustered a set of N data points using two different specified clustering algorithms. In both cases we obtained 5 clusters and in both cases the centers of the clusters are exactly the same. Can 3 points that are assigned to different clusters in one method be assigned to the same cluster in the other method?
3. What are the characterics of noise points in DBSCAN?

#### M.20.2.4 Section 4

**Section title:** Deep Learning

**Topics covered in this section:**

- Artificial Neural Networks
- Back-propagation
- Convolutional Neural Networks
- Autoencoder
- Variatonal Autoencoder
- Generative Adversairal Networks

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is a fully connected feed-forward ANN?
2. Explain different hyperparameters of CNNs.
3. Calculate KL-divergence between two probability distributions.
4. What is a generative model and how is it different from a discriminative model?

**Typical questions for seminar classes (labs) within this section**

1. Implement different types of ANNs to solve to solve different classification problems.
2. Calculate KL-divergence between two probability distributions.
3. Implement different generative models for different problems.

**Test questions for final assessment in this section**

1. Explain what is ReLU, what are its different variants, and what are their pros and cons?
2. Calculate the number of parameters to be learned during training in a CNN, given all important information.
3. Explain how a VAE can be used as a generative model.

## M.21 Managing Software Development

- **Course name:** Managing Software Development
- **Course number:** SE-04

### M.21.1 Course characteristics

#### M.21.1.1 Key concepts of the class

- People Management
- Processes and Software Development Life-cycles
- Planning and Controlling

#### M.21.1.2 What is the purpose of this course?

Large scale software development requires the ability to manage resources - both human and computational - through control of the development process. This course is a breadth oriented course, designed to help technically-trained software engineers to acquire the knowledge and skills necessary to lead a project team, understand the relationship of software development to overall project engineering, estimate time and costs, and understand the software process. The nature of software development is sufficiently unique to require specialized management techniques, especially in the areas of the estimating and scheduling.

#### M.21.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students will remember:

- importance of people in software development
- trade-offs in various managing style
- importance to manage customer expectations
- major differences in the software development process
- several methods to plan projects and estimate their duration

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- Impact of different personality types on team performance
- Strengths and weaknesses of specific software development processes
- Ways to define the project success criteria and potential risks
- Customer oriented work breakdown and planning

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Apply critical reasoning to assess root-causes of problems in software development process
- Select a right process of a particular software development project
- Define the success thresholds.
- Evaluate and manage risks
- Estimate project duration with several methods
- Communicate with customers on the project related matters.

#### M.21.1.4 Course evaluation

## M.21.2 Evaluation

**Table M.54:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 10 |
| Interim performance assessment | 30 | 90 |
| Exams | 50 | 0 |

The students performance will be evaluated as follows:

- Participation (10%)
- Case study analysis and assignments (60%)

- Reading questions (10%)
- Final paper (20%)

#### M.21.2.1 Grades range

**Table M.55:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 65-79 |
| C. Satisfactory | 60-74 | 50-64 |
| D. Poor | 0-59 | 0-49 |

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table M.73, but it may change slightly (usually reduced) depending on how the semester progresses.

#### M.21.2.2 Resources and reference material

- This course makes use of many reference materials that are posted to Moodle.
-
-

### M.21.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.56:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | People Management | 14 |
| 2 | Software development processes | 12 |
| 3 | Defining project success criteria | 10 |
| 4 | Planning and controlling software development projects | 16 |

#### M.21.3.1 Section 1

**Section title:**

People Management

**Topics covered in this section:**

- MSD Introduction
- OBU Case Study Discussion
- Managing Technical People
- Team formation, Decision Making and Conflict Resolution
- Managing Customer Expectations
- MCE Case Study Discussion

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Simply list the separate processes that a development team should manage. Also explain why each is, or is not important?
2. What are typical personality treats of "technical people"?
3. What are the Tuckman's team development stages?
4. What leadership style(s), as discussed by Goleman, would be counterproductive in certain cases?
5. Looking at McCarthy's article on participatory leadership, in what situation would a "coercive" style be used in team building?
6. The Heskett paper effectively recommends involving customers more in your work. What are some software development activities that would be suited to this? What are some where you don't think customers should be involved and why?

**Typical questions for seminar classes (labs) within this section**

1. Based on OBU case, what were the top issues involved with the development of MS Word with regards to stakeholders, market, technology, business drivers, culture and schedule?
2. Apply leadership/teamwork concepts to a project development problem at SATERA.
3. Discuss the key issues dealing with managing customer expectations with the Gigaplex client and propose a course or courses of action to resolve the problems you identified.

**Test questions for final assessment in this section**

1. Based on a brief description of a situation on a project, give a recommendation for a leadership style. Justify.
2. Name Tuckman's team development stages? What activities are typical of each of those stages?
3. Explain the Barry Boehm's few on managing customers expectations. Give an example.

#### M.21.3.2 Section 2

**Section title:**

Software development processes

**Topics covered in this section:**

- Defining and measuring processes
- Case Study Exercise Discussion
- Software Development Life-cycles
- Process Frameworks and How to choose

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Pressman identifies some misuse of process metrics. Which of these misuses is the most harmful? Explain why.

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

2. Pressman states that “correctness” is “the degree to which the software program performs its required function.” He says that defects per KLOC is the most common measurement for correctness. Do you agree or disagree that this is an appropriate measurement and why?
3. What do you think are some negatives to using Goal Question Metrics?
4. What is the difference between a lifecycle and a process framework? Give some specifics.
5. What assumptions should you make about stakeholders when choosing an incremental model? Explain why you chose these.
6. Is there a difference between iterative and incremental models? Explain.
7. What lifecycle does the ACDM framework fit under? Explain.

**Typical questions for seminar classes (labs) within this section**

1. Specifically identify three current (time paper was written) critical Tartan Case software development processes.
2. In Tartan Case, what metrics are collected, or should be collected with an eye to improving the processes. Include a discussion on analyzing and changing the processes.
3. Provide justified recommendations to improve Tartan’s software quality. Specifically, what would you change and what metrics would you use to analyze the effectiveness of current processes and/or recommended changes?
4. Determine what SDLC and process framework you would recommend and why for each of the four software projects: OBU, Denver Baggage, Avionics development, and Department of Transportation.

**Test questions for final assessment in this section**

1. Give definition of a process.
2. What is a Software Development Life-Cycle and how does it differ from from a process?
3. List several software development processes currently in use and give their trade-offs?
4. Based on a brief description of a software project, what software development process would you recommend? Justify.

### M.21.3.3 Section 3

**Section title:**

Defining project success criteria

**Topics covered in this section:**

- Requirements Management
- Case Study Exercise Discussion
- Requirements case study/lecture
- Identifying and Managing Software Risk
- TBQ, Threshold of Success
- Risk statement discussion

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the difference between verifying and validating requirements? Be specific.
2. What are the main activities of requirements engineering?

3. What are the requirements elicitation problems?
4. What are the Quality Attributes? Describe ways to elicit quality attributes requirements.
5. Explain how one can use the Pareto principle in the context of risk management. Give an example.
6. Give examples of methods to assess risk impact.
7. What is the difference between Threshold of Success and Risks?

**Typical questions for seminar classes (labs) within this section**

1. Describe the key issues that found in the case study with regards to requirements management.
2. Explain what approaches are selected in the case study and why? What was the result of applying those approaches on the project and team how addressed these issues?
3. According to the case study, what are the most critical factor in software quality and the most difficult problem that that the team faced in their projects?
4. Identify 5 key risks in case study and prioritize them. Simply state and justify your prioritization method. Ensure these are in the Condition-Consequence format, include context statements and time horizons.
5. Provide mitigation strategies to the risk identified in the case study.
6. Develop Threshold of Success for the case study project.

**Test questions for final assessment in this section**

1. Given a situation, identify, prioritise possible risks and discuss mitigation strategy.
2. What is Threshold of Success and hos does it differ from Risks.
3. Explain the condition-consequence definition for risk statements.
4. Give an example of criticality assessment for risk statements.

#### M.21.3.4 Section 4

**Section title:**

Planning and controlling software development projects

**Topics covered in this section:**

- Introduction - Planning & Controlling Software Development Projects
- Work Breakdown Structures
- Estimation Methods
- Activity Planning
- Milestone Planning
- Release Planning
- Tracking Reporting & Controlling

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is Work Breakdown structure and how can it be applyed for project planning?
2. List several estimation methods and discuss their trade-offs?
3. What is the activity planning?
4. Give a definition of a Critical Path?
5. What is milestone planning?
6. Discuss several methods for release planning?

**Typical questions for seminar classes (labs) within this section**

1. Propose a Work Breakdown Structure for a given project.
2. Give minimum duration of a project give a list of tasks and their dependencies.

3. Develop a milestone plan.
4. Propose a release plan for a given project based on estimations and priorities of tasks.

**Test questions for final assessment in this section**

1. List several types of Work Breakdown Structures and discuss trade-offs.
2. Apply critical path calculations to an activity graph. Give the latest start for a specific task.
3. Define milestone planning.
4. Explain MOSCOW method for release planning.
5. During semester study and write a paper on a software engineering topic of your choice.

# M.22 Metrics and Empirical Methods for Software Engineers and Data Scientists

- **Course name:** Metrics and Empirical Methods for Software Engineers and Data Scientists
- **Course number:** XYZ

## M.22.1 Course Characteristics

### M.22.1.1 Key concepts of the class

- Goal-Question-Metric approach
- Basics of statistics
- Measurement and metrics in software development

### M.22.1.2 What is the purpose of this course?

The main purpose of this course is to present the fundamentals of metrics and empirical methods to the future software engineers and data scientists, on one side providing the scientific fundamentals of the disciplines, and on the other anchoring the theoretical concepts on practices coming from the world of software development and engineering. As a side product, the course also refreshes the fundamentals of statistics, providing the basis for more advanced statistical courses in the following semester(s) of study.

### M.22.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to:

- Remember the fundamentals of statistics and probability theory
- Remember the specifics and purpose of different measurement scales
- Distinguish between random variable and random process
- Explain the difference between the correlation and causation
- Remember existing object-oriented, functional and quality metrics

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to understand:

- the value of measurement for software engineers and data scientists
- the basic concepts of measurement
- the concept of correlation
- the fundamental laws in statistics
- the concept of Goal-Question-Metric approach

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

- Apply Goal-Question-Metric approach in practice
- Apply statistics and probability theory in practice
- Apply hypothesis testing technique in software analysis
- Analyze software development process using fundamental size and complexity measures
- Apply static code analysis using object oriented metrics

### M.22.1.4 Course evaluation

**Table M.57:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Weekly quizzes | ? | 20 |
| Personal GQM | ? | 20 |
| Midterm | ? | 20 |
| Final exam | ? | 40 |

#### M.22.1.5 Grades range

**Table M.58:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 95-100 |
| B. Good | 75-89 | 75-94 |
| C. Satisfactory | 60-74 | 55-74 |
| D. Poor | 0-59 | 0-54 |

#### M.22.1.6 Resources and reference material


- Norman Fenton and Shari Lawrence Pfleeger. Software Metrics: A Rigorous and Practical Approach. International Thomson Computer Press, London, UK, 2nd edition, 1997
- Donald T. Campbell and Julian C. Stanley. Experimental and Quasi-Experimental Designs for Research. Rand McNally College Publishing, 1963
- Larry Wasserman. All of Statistics: A Concise Course in Statistical Inference. Springer Texts in Statistics. Springer, New York, 2004. ISBN 978-1-4419-2322-6. doi: 10.1007/978-0-387-21736-9
- Oliver Laitenberger and Dieter Rombach. Lecture Notes on Empirical Software Engineering. chapter (Quasi) Experimental Studies in Industrial Settings, pages 167–227. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2003. ISBN 981-02-4914-4
- Rini van Solingen and Egon Berghout. The Goal/Question/Metric Method: a practical guide for quality improvement of software development. The McGraw-Hill Companies, Cambridge, England, 1999. ISBN 077-709553-7.
- Andrea Janes and Giancarlo Succi. Lean Software Development in Action. Springer, Heidelberg, Germany, 2014. ISBN 978-3-662-44178-7. doi: 10.1007/978-3-642-00503-9

## M.22.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.59:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Concept of measuring | 12 |
| 2 | Fundamentals of statistics | 24 |
| 3 | Measurement in software development | 12 |

### M.22.2.1 Section 1

**Section title:** Concept of measuring

**Topics covered in this section:**

- Measurement: concept, definition and fundamentals of measurement
- Goal-Question-Metric approach
- Representational theory of measurement
- Measurement scales and functions that can be applied to scales

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are the phases of GQM? How are they connected to each other? What are steps of GQM method?
2. What does SWOT mean?
3. What is the measurement?
4. How measurement can help us to understand, control and improve development process??

5. What does Representation Condition mean?
6. What is the Measurement Scale?
7. What are characteristics of a good measurement? What is the difference between validity and reliability?

**Typical questions for seminar classes (labs) within this section**

1. Which benefits the GQM provides to you as a Software Engineer / Data Scientist?
2. Imagine your goal is to ”increase availability of some software system”. Provide Questions and Metrics for this goal.
3. What is measurement Reliability and measurement Validity? What are the differences between the two? Provide an example of reliable, but invalid measurement and an example of valid, but unreliable measurement
4. Which Measurement Scales do you know? What are the differences between them? Provide examples for each of them.
5. Provide an example of Representation Condition
6. Create metrics that measures your study progress, outline the properties of such metrics in terms of subjective vs. objective, direct vs. indirect, etc; detail how you will collect your metrics, concretely and check your metric on reliability & validity

**Test questions for final assessment in this section**

### M.22.2.2 Section 2

**Section title:** Fundamentals of statistics

**Topics covered in this section:**

- Basic concepts of probability theory
- Random variable and random process
- Linear regression
- Correlation and convolution
- Moments and moment generating functions
- Law of Large Numbers
- Central Limit Theorem

- Hypothesis testing

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Describe the three approaches to probability.
2. Write the fundamental theorem of algebra.
3. Write the general structure of the OLS equation for one variable.
4. Write the general structure of the OLS equation for the case of multiple independent variables.
5. What is the connection between the correlation coefficient and the coefficient of determination? What does each of them show?
6. State and prove the Law of Large Numbers
7. State and prove the Central Limit Theorem
8. Explain what are H0 and H1 in hypothesis testing
9. Explain the role of the Bonferroni inequality in hypothesis testing

**Typical questions for seminar classes (labs) within this section**

1. Define and provide examples of sample space, events and probability measure
2. Write the formula for the coefficients of the simple linear regression. Explain the mathematical procedure you do to derive them and derive them
3. Fully deduce the value of the coefficient in OLS equation for multiple independent variables
4. Calculate the correlation between two functions and explain its meaning
5. Calculate the Pearson coefficient for the given functions
6. Write and prove Markov's inequality. Write and prove Chebyshev's inequality. How these theorems related to the LLN?

7. Deduce the MGF for normal distribution
8. Provide a concrete example of a test, detailing both H0 and H1
9. State and prove the Bonferroni inequality

**Test questions for final assessment in this section**

### M.22.2.3 Section 3

**Section title:** Measurement in software development

**Topics covered in this section:**

- Fundamental software measures of size and complexity
- Halstead metrics
- Object-oriented metrics
- Function points
- Quality metrics

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
| --- | --- |
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are advantages and disadvantages of LOC measure?
2. How we can measure the complexity of the given program module?
3. What is Halstead metrics, their strength and weaknesses?
4. What is Chidamber & Kemerer metrics suite?
5. What is the purpose of high cohesion and low coupling in object-oriented programming?
6. What are function points and functional size measurement?
7. What is Mk II functional point analysis?
8. Which activities are included in the effort estimation of a software project?

9. What is product and process quality metrics? What is the difference between them?

**Typical questions for seminar classes (labs) within this section**

1. For the given example of code compute its LOC, FAN IN and FAN OUT metrics
2. Analyze cyclomatic complexity of the given code using 4 different approaches? Which one is the easiest for you and why?
3. For the given example of code compute its Halstead metrics
4. For the given example of code compute its CK metrics?
5. Describe the benefits of Function Point Analysis.
6. Describe the difference between Mk II FPA and Albrecht's FPA.

### M.22.2.4 Test questions for final assessment in the course

1. State the Fenton Measurement theory and explain what is Representation condition?
2. Define the steps needed to elaborate a GQM
3. Describe the Taylor
4. Describe the fundamental theorem of Algebra
5. Based on the concept the Taylor theorem and the fundamental theorem of Algebra, explain whether the number of datapoints should be equal, smaller or larger than the number of independent variables (features) and why
6. Explain how the GQM can be used to define appropriate number of variables
7. For the given function compute its mean, mode, median, standard deviation
8. Define the OLS linear regression in the case of one and multiple variables and deduce their parameters
9. For the given two functions compute their Covariance, Pearson's correlation coefficient and describe results
10. State and prove weak and strong formulation of LLN
11. State Lindeberg–Lévy formulation of CLT and prove it
12. Compute LOC, MCC, FI, FO for given code. Describe how to apply MCC metrics and meaning of FI-FO output
13. For the given module compute the CK metrics for its classes

# M.23 Models of Software Systems

- **Course name:** Models of Software Systems
- **Course number:** SE-07

## M.23.1 Course characteristics

### M.23.1.1 Key concepts of the class

- Formal Models
- Software Specification
- Software Verification

### M.23.1.2 What is the purpose of this course?

Formal models and formal methods are gaining an increasing importance today - in academia and in industry - in particular due to the emerging of tool support used for modeling and analysis of systems. An important question is asked world-wide when developing academic curricula: what a software engineer should know about formal modeling? On what specific aspect a course should focus and help the perspective engineer to navigate such a complex and fast changing field? The course attempt to give a partial although consistent answer to this question by providing an introduction to the foundations and uses of formal modeling. The course can be seen as organized in three parts: the first part sets out the mathematical foundations on which the rest of the course is based. It presents the mathematical concepts, such as logics, proofs, theories, sets, functions, relations, etc., on which all modern modeling approaches build. The second part presents the concepts that allow to relate mathematics to computation and to software: state machines, invariants, pre- and post-conditions, proving properties about programs. The third part consists of particular notations and method of formal modeling showing how the general concepts of the second part can be turned into effective engineering tools.

### M.23.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define

- Define mathematical notions behind software specification and verification
- List different specification methods
- List different specification notations
- Define concurrency and related concepts
- List Popular verification tools

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples)

- Describe the basic mathematical machinery and how can be applied to improve software development and software quality
- Explain Strengths and weaknesses of specific models and logics
- Explain State machines and temporal logics
- Abstract formal models for certain classes of systems

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply

- Formally modelling a system
- Reasoning about system properties
- Specifying a system in Z
- Coding in languages like PROMELA for model checking
- Using SPIN model checker

### M.23.1.4 Course evaluation

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

**Table M.60:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 40 |
| Interim performance assessment | 30 | 30 |
| Exams | 50 | 30 |

#### M.23.1.5 Grades range

**Table M.61:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 65-79 |
| C. Satisfactory | 60-74 | 50-64 |
| D. Poor | 0-59 | 0-49 |

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table M.73, but it may change slightly (usually reduced) depending on how the semester progresses.

#### M.23.1.6 Resources and reference material

- Handouts supplied by the instructor
-
-
-
-
-
-

### M.23.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

Table M.62: Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Mathematical Foundations | 12 |
| 2 | Concepts of computation and software | 8 |
| 3 | Formal Modelling | 32 |

### M.23.2.1 Section 1

#### Section title:

Mathematical Foundations

#### Topics covered in this section:

- Propositional & Predicate Logic
- Sets, Relations, Functions

#### What forms of evaluation were used to test students' performance in this section?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

#### Typical questions for ongoing performance evaluation within this section

1. What is a formal system? What is a rewriting system?
2. What is the difference between propositional and predicate logic?
3. What is a predicate? What is a connective?
4. What is a quantifier? Existential? Universal?
5. Given a specific formula define the truth table?
6. Check if a specific formula is a tautology or a contradiction.
7. State the difference between relation and function
8. Computing the intersection or union of two sets
9. What are De Morgan Laws? Example of applications.
10. Operators of set theory.

**Typical questions for seminar classes (labs) within this section**

1. Check if given formulas are well-formed. For those that are not briefly explain why.
2. Prove that a given statement is a theorem in a given formal system.
3. Augment a given formal system so that given statements become theorems in that formal system.
4. Construct a truth table for given propositions.
5. For given satisfiable propositions, provide z3 code that proves their satisfiability.
6. Translate given predicates into English sentences using the provided translation key.
7. Translate given sentences into predicate logic, using the translation key provided.
8. Identify theorems among given propositional statements using z3.
9. Provide a linear/tree proof of a statement identified as a theorem.
10. Prove in Isabelle proof assistant a statement identified as a theorem.

**Test questions for final assessment in this section**

1. Given a set of different formulas define truth tables
2. Given a set of different formulas define those that are tautology and/or contradiction
3. Compute operations over some specific instances of sets
4. Determining satisfiability of a formula
5. Model natural language as propositional or predicate logic formulas

#### M.23.2.2 Section 2

**Section title:**

Concepts of computation and software

**Topics covered in this section:**

- Proof Techniques, Induction
- State Machines

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 1 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is a state machine and how it is formally defined?
2. What is s state, initial state, final state?
3. What is the language accepted by a machine?
4. Design a state machine for some specific purpose
5. Verify that a state machine performs a given goal
6. How does a proof system work?
7. What is natural deduction and for what it is used?
8. What is an inference rule?
9. Prove specific theorems in natural deduction
10. What are preconditions and postconditons?

**Typical questions for seminar classes (labs) within this section**

1. Give a 4-tuple description of a state machine.
2. What are the reachable states of a state machine?
3. Implement a state machine in the LTSA tool; provide your code.
4. Does a state machine have any progress or safety violations? Provide the corresponding output from LTSA. .
5. Given a UML class diagram, annotate its classes with class invariants in OCL.
6. Given a UML class diagram, annotate its methods with preconditions in OCL.
7. Given a UML class diagram, annotate its methods with postconditions in OCL.

**Test questions for final assessment in this section**

1. Prove theorems in natural deduction
2. Design a state machine for some specific purpose

3. Verify that a state machine performs a given goal
4. Determine the language recognized by a state machine

### M.23.2.3 Section 3

**Section title:**

Formal Modelling

**Topics covered in this section:**

- UML
- Z/B/Event-B
- Model Checking
- Linear Temporal Logic
- Concurrency
- Petri Nets
- Alloy
- Process Algebra
- CSP
- Autoproof

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are the UML diagrams and what is their purpose?
2. What is OCL and what is its purpose?
3. What is Z and what is its purpose?
4. What is Design-by-contract and what is its purpose?
5. Design UML Use Cases diagrams for some given system

6. Design UML Class Diagrams for some given system
7. Define OCL constraints for some parts of some given system
8. Specify system requirements for a given system in Z
9. Use temporal logic and model checkers to specify and verify system properties
10. Use Design-by-contract and Autoproof to specify and verify system properties

**Typical questions for seminar classes (labs) within this section**

1. Specify a state space in Z.
2. Specify initial states of a state space in Z.
3. Specify CRUD operations of a state space in Z.
4. Define a robust version of a Z specification.
5. Provide an Alloy specification of a state space.
6. Enrich an Alloy specification with axioms.
7. Implement CRUD operations on an Alloy specification of a state space.
8. Specify an assertion that you want to hold on an Alloy specification.
9. Check an assertion in an Alloy specification using Alloy analyzer; correct the specification if the assertion does not hold.

**Test questions for final assessment in this section**

1. Design UML Use Cases diagrams for some given system
2. Design UML Class Diagrams for some given system
3. Define OCL constraints for some parts of some given system
4. Specify system requirements for a given system in Z
5. Use temporal logic and model checkers to specify and verify system properties
6. Use Design-by-contract and Autoproof to specify and verify system properties

# M.24 Neuroscience

- **Course name:** Neuroscience
- **Course number:** ???

## M.24.1 Course Characteristics

### M.24.1.1 Key concepts of the class

- An introduction to modern neuroscience and neural engineering.
- Basic principles of nervous system functioning and its applications for rehabilitation, robotics and control of human brain states.

### M.24.1.2 What is the purpose of this course?

This course is designed to serve an introduction to modern neuroscience and neurotechnologies. Because neuroscience is an interdisciplinary field including biological science, physics, mathematics, philosophy, and engineering, we will cover all the areas in lectures and seminars. The course goal is to make students familiar with basic principles of nervous system functioning and its applications for rehabilitation, robotics and control of human brain states. The students will learn about various methods of neuron modelling and biological bases of artificial intelligence. We will discussed the various neuroimaging methods and its applications for disease diagnostics and brain computer interfaces development. After the course the students will be able to conceptually understand the important terms and approaches of modern neuroscience and neural engineering. The theoretical introduction will be complemented by practical examples of task solving.

### M.24.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and differentiate

- Physiology and functions of neurons,
- Basic principles of realization of human cognitive abilities in the brain,

- Structure and functions of brain,
- Main brain diseases and methods of neurorehabilitation,
- Main principles of brain-computer and brain-machine interfaces.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be

- able to describe what modern neuroscience is and what a neurotechnologies might include or not include, and why, to someone not in the class
- learn about many of the features of the nervous system and body that may be useful to a specialist in artificial intelligence,
- familiar with modern computational neuroscience methods and approaches,
- learn about some of the goals of neuronal engineering and neurotechnologies and who is involved (in research and uses)

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

- Know how to construct brain-computer interfaces,
- Simulate spiking neuronal network,
- Know how to choose the neuroimaging technique depending on the required information,
- Classify different types of brain activity.

#### M.24.1.4 Course evaluation

**Table M.63:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 50 |
| Interim performance assessment | 30 | 30 |
| Exams | 50 | 20 |

If necessary, please indicate freely your course's features in terms of students' performance assessment:

#### M.24.1.5 Grades range

**Table M.64:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 65-79 |
| C. Satisfactory | 60-74 | 50-64 |
| D. Poor | 0-59 | 0-49 |

If necessary, please indicate freely your course's grading features: The course grades were given according to the following rules: Lab projects (2) = 50 pts, Quizzes (3) = 30 pts, Exam (1) = 20 pts.

The subject of the course is far from the materials previously studied by the students, so in many lectures it is completely new to them. Therefore the grade requirements for this course are relaxed

#### M.24.1.6 Resources and reference material

The course is build based on these main textbooks:

- Fundamental Neuroscience (3rd edition), Academic Press, Elsevier, 2008
- Galizia, C. Giovanni, and Pierre-Marie L. (ed.) Neurosciences-From Molecule to Behavior: an University Textbook. Springer Spektrum, 2013.

Other reference material:

- Baars, B. J., and Nicole M. G. Cognition, brain, and consciousness: Introduction to cognitive neuroscience. Academic Press, 2010.
- Nam C. S., Nijholt A., Lotte F. (ed.) Brain-Computer Interfaces Handbook: Technological and Theoretical Advances. CRC Press, 2018. .
- The review papers from Nature Neuroscience; https://www.nature.com/neuro/.

### M.24.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.65:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction in Neuroscience | 2 |
| 2 | Computational neuroscience | 20 |
| 3 | Neuroanatomy and functions of brain | 14 |
| 4 | Cognitive neuroscience | 10 |
| 5 | Brain-computer interfaces | 10 |
| 6 | Brain diseases and neurorehabilitation | 4 |

#### M.24.2.1 Section 1

**Section title:** Introduction in Neuroscience.

**Topics covered in this section:**

- Subject of neuroscience
- History of neuroscience

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Who got the Nobel Prize for extensive observations, descriptions, and categorizations of neurons throughout the brain? At what year?
2. Who got the Nobel Prize for discoveries regarding the functions of neurons? At what year?
3. Who and when discovered the effect of the electrical excitability of muscles and neurons?
4. Draw and explain the scheme for measuring the speed of an impulse along a nerve fiber proposed by Hermann von Helmholtz.

**Test questions for final assessment in this section**

1. What is a subject of neuroscience?

2. When and by whom was the neural doctrine proposed?
3. What is the essence of optogenetic imaging of neurons?

#### M.24.2.2 Section 2

**Section title:** Computational neuroscience

**Topics covered in this section:**

- Functional classes of neurons
- Basic function of neurons
- Structural classes of neurons
- Glial cells
- Neuron communications and action potential
- Basic concepts of computational neuroscience
- Turning curves
- Plasticity in neuronal systems
- Mathematical model of action potential generation

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. The connection between one of the two neurons decreased by 20%, and between the other increases by 20% due to STDP. Qualitatively draw the generation of spikes of both pairs of neurons. Explain your answer.
2. Identify the type(s) or subtype(s) of neural cell performing the following functions:
   - Connecting neurons within specific regions of the central nervous system
   - Forming myelin in the motor neurons

- Providing the appropriate ionic milieu for neurons to be able to generate action potentials

3. Draw and briefly explain voltage-clamp experiment design (aim, scheme, idea, result).
4. The persistent current through neuron membrane is determined by:
   - all channels;
   - channels that do not have inactivation gates
   - channels that have inactivation gates
   - channels that do not have activation gates
   - channels that have activation gates

**Typical questions for seminar classes (labs) within this section**

f

1. Solve numerically the system of differential equations describing the Hodgkin-Huxley neuron model using the Runge Kutta 4th order method, analyze the correctness of choosing the time step.
2. Analyze different regimes of a neuron dynamics, plot time series and phase portraits of the signal, calculate a regime map.
3. Solve numerically the system of stochastic differential equations corresponding to the Hodgkin-Huxley neuron model with noise, analyze the influence of noise amplitude on the system dynamics.
4. Solve numerically the system of differential equations describing 2 noisy Hodgkin-Huxley neurons coupled by chemical synapses, analyze synchronization between neurons for different values of the coupling strength.
5. Numerical simulation of network of 10 Hodgkin-Huxley neurons with global topology.
6. Analyze the influence of external stimulus and noise amplitude by calculating characteristic correlation time.

**Test questions for final assessment in this section**

1. What are the functions of glial cells?

2. Which of the following best characterizes an action potential that occurs in a neuron?
   - Sometimes strong
   - All-or-none
   - Very rarely weak
   - Sometimes weak
3. Adaptation of a neural response to a constant stimulus? (mark all correct answers):
   - leads to decrease amplitude of spikes
   - lleads to increase neuron firing rate
   - lleads to decrease neuron firing rate
   - lleads to increase amplitude of spikes
   - lremoves the one-to-one relationship between firing rate and magnitude of the stimulus
   - lremoves the one-to-one relationship between amplitude of spikes and magnitude of the stimulus

#### M.24.2.3 Section 3

**Section title:** Neuroanatomy and functions of brain

**Topics covered in this section:**

- Central and peripheral nerves systems
- Autonomic Nervous System
- Brain organization
- Invasive and non-invasive neuroimaging technics
- Structural brain connectivity
- Functional brain connectivity
- Introduction in graph theory
- A taxonomy of methods for functional connectivity detection
- Model-based and model-free methods for functional connectivity restoration
- Non-directed and directed methods for functional connectivity restoration

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 01 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Which of the following functional connectivity recovery methods are model-free ones?
   - Recurrence measure of dependence
   - Mutual information
   - Granger causality
   - Phase locking index
   - Pierson correlation coefficient
2. The BOLD signal measures...
   - the diffusion of water molecules in the brain
   - the magnetic field excited by the post-synaptic current flow along the dendrites of pyramidal nerve cells
   - the local changes in blood oxygenation ocurring during brain activity
   - the spike activity of single nerve cells
   - the activation of various neurotransmitters during brain activity
3. How can you characterize the rich club organization pattern in network?
4. Which part of the brain is important for language comprehension?

**Typical questions for seminar classes (labs) within this section**

1. Use Pearson correlation and recurrence measure of dependence to estimate the connections between different EEG channels in $\alpha$ (8-12 Hz) and $\beta$ (15-30 Hz) frequency regions for different values of brightness intensity $I \in (0.1 - 1.0)$, find an optimal value of $I$ for each subject characterized by maximal connectivity
2. Estimate the efficiency of Pearson correlation and recurrence measure of dependence.

3. Investigate the time-dependence of the coupling strength, find an optimal value of time window for calculation of the coupling strength.
4. Use a Nonparametric Test for checking a hypothesis of achieving maximal connectivity in the found range of the optimal brightness intensity.

**Test questions for final assessment in this section**

1. Describe the concept of the structural and functional network of the brain. Give examples.
2. For a given functional network described by graph, calculate degree of nodes, mean shortest path, centrality.
3. Describe pros and cons of fMRI neuroimaging technics.

### M.24.2.4 Section 4

**Section title:** Cognitive neuroscience

**Topics covered in this section:**

- Neurophysiology of cognitive processes
- Brief historical tour in cognitive neuroscience
- Basic cognitive processes: sensations
- Basic cognitive processes: perceptions
- Basic cognitive processes: attention
- Basic cognitive processes: memory

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is Stroop effect? Repeat the Stroop's experiment yourself

2. Draw the scheme of analysis of visual information in eye.
3. What is the difference in the processing of visual information passing through the ventral and dorsal pathways in the brain?
4. Consider the methods of cognitive neuroscience.

**Test questions for final assessment in this section**

1. Describe the role of the thalamo-cortical network in sensory processing.
2. What are the basic cognitive abilities you know?
3. What types of human sensation do you know?

### M.24.2.5 Section 5

**Section title:** Brain-computer interfaces

**Topics covered in this section:**

- Classification of brain-computer interfaces (BCI)
- EEG preprocessing methods for BCIs
- Brain activity pattern recognition and classification in multichannel data
- BCI applications

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Name brain activity patterns for control commands for BCI
2. Describe the functional model of reactive BCI
3. Describe the speller based on P300 potential recognition
4. Describe the process of generation and detection of sensorimotor rhythm

**Typical questions for seminar classes (labs) within this section**

1. Create an artificial neural network with 31 input and 1 output neurons to work with EEG data.
2. Train the artificial neural network for recognition of EEG patterns corresponding to the human's movement of left or right leg by half of the data.
3. Apply the trained artificial neural network to the rest of the data.
4. Estimate the accuracy of the recognition.

**Test questions for final assessment in this section**

1. Describe typical functional model of BCI.
2. What is difference between active and passive BCIs?.
3. Which of the following is a event-related potential?
    - Cortical potential
    - P300 evoked potential
    - Event-related synchronization/desynchronization (ERS/ERD)
    - Eye movement artefact

### M.24.2.6 Section 6

**Section title:** Brain diseases and neurorehabilitation

**Topics covered in this section:**

- Types of nervous diseases
- Epilepce
- Neurodegenerative diseases
- Stroke and impaired motor function
- Neurorehabilitation

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. Is it possible to use the brain-computer interface to treat epilepsy?
2. What are the main features of Parkinson's disease?
3. Describe the role of modern neuroimaging techniques in the diagnosis of nervous diseases
4. Describe the main neurorehabilitation methods.

| Form | Yes/No |
| --- | --- |
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

### Test questions for final assessment in this section

1. How can epilepsy be diagnosed?
2. Name pros and cons of exoskeleton-based rehabilitation after stroke.
3. What are the main features of Alzheimer's disease?

# M.25 Offensive Technologies

- Course name: Offensive Technology
- Course number: ?

## M.25.1 Course characteristics

### M.25.1.1 Key concepts of the class

- Physical Security
- Network security
- Web Security
- Software Security
- GSM

### M.25.1.2 What is the purpose of this course?

Offensive Technology introduces methods and tools to the students to assess the security of different services and protocols therein. The course aims to expose the students to real-world problems from a security point of view and let them find vulnerabilities in both software and hardware. The course hosts exciting and interesting topics. Furthermore, the students will develop projects of their choice to show their skills. In this course the students will particularly focus on physical security, network security, web injection flaws, advanced memory exploit/mitigation and fuzzing techniques.

### M.25.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?** By the end of the course, the students should be able to recognize and define

- RFID
- A rogue Wifi access point
- ARP & DNS spoofing
- Sniffing & MAC flooding
- Covert channel

- Honey pots
- SSL meet in the middle
- Common weaknesses/vulnerabilities in web application
- ASLR, NX and how are these techniques can help to protect against a malicious attacker
- Fuzzing techniques
- 2G, UMTS, LTE, LTE-A, ARFCNs, and hopping channels

**- What should a student be able to understand at the end of the course?** By the end of the course, the students should be able to describe and explain (with examples)

- Vulnerabilities on Mifare Classic
- Ways to get in a passive eavesdropper position
- Ways to get in an active intruder position
- Ways to bypass memory mitigation techniques
- The vulnerability of the A5/1 stream cipher

**- What should a student be able to apply at the end of the course?**

- Clone HID and Mifare ID Tags
- Setup port-mirroring and IDS
- Setup an inline mode IPS
- Perform a network discovery
- Be in the middle by various means of spoofing
- Be in the middle by various means of rogue services
- Proceed with SSL/TLS meet in the middle attack
- Detect/exploit common weaknesses/vulnerabilities in web applications.
- Detect vulnerabilities in software.
- Writing an exploit to bypass ASLR and NX protection.
- Perform fuzzing for a specific use case.
- Evesdrop on 2G SMS and voice calls

Table M.66: Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | ? | 80 |
| Exams | ? | 20 |

#### M.25.1.4 Course evaluation

If necessary, please indicate freely your course's features in terms of students' performance assessment: The laboratory assessments are particularly taken care of, and the tasks do correspond with the teachings from the lectures.

#### M.25.1.5 Grades range

Table M.67: Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 90-100 |
| B. Good | 75-89 | 70-89 |
| C. Satisfactory | 60-74 | 60-69 |
| D. Poor | 0-59 | 0-59 |

If necessary, please indicate freely your course's grading features: The laboratory assignments are mandatory with a requried minimum result of 6/10 - including retakes - to complete the course. As a conscequence, the grades are generally pretty high and therefore the grading ranges are scaled up.

#### M.25.1.6 Resources and reference material

- Mike O'Leary, Cyber Operations, Second Edition, Apress, 2019
- Michal Zalewsk, The Tangled Web, No Starch Press, 2011
- Jon Erickson, Hacking: The Art of Exploitation, 2nd Edition, No Starch Press, 2008
- The Fuzzing Book `https://www.fuzzingbook.org`
- Jörg Eberspächer, Hans-Joerg Vögel, Christian Bettstetter, Christian Hartmann, GSM - Architecture, Protocols and Services, Third Edition, Wiley, 2009

## M.25.2 Course Sections

**Table M.68:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Physical & Network Security | 8 |
| 2 | Web Security | 4 |
| 3 | Software Security | 8 |
| 4 | GSM | 4 |
| 5 | Labs | 56 |

### M.25.2.1 Section 1

**Section title: Physical & Network Security**

**Topics covered in this section:**

- Physical access
- RFID
- Sniffing & network discovery
- MAC flooding
- ARP spoofing
- DNS spoofing
- Rogue DHCP
- Rogue Wifi Access Point
- Port mirroring & Intrusion Detection Systems
- Intrusion Prevention Systems
- Covert channels
- Honey pots
- SSL meet in the middle

**What forms of evaluation were used to test students' performance in this section?**

**Typical questions for ongoing performance evaluation within this section**

- What is RFID?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

- What is a rogue Wifi access point?
- What is ARP & DNS spoofing?
- What is sniffing & MAC flooding?
- What is a covert channel?
- What is a honey pot?

**Typical questions for seminar classes (labs) within this section**

- Clone HID and Mifare ID Tags
- Setup port-mirroring and IDS
- Setup an inline mode IPS
- Perform a network discovery
- Be in the middle by various means of spoofing
- Be in the middle by various means of rogue services
- Proceed with SSL/TLS meet in the middle attack

**Test questions for final assessment in this section**

- Briefly describe one Mifare classic weakness and an afferent attack
- By what means can you get in a passive eavesdropper position?
- By what means can you get in an active intruder position?
- Describe different methods to perform ARP spoofing in term of both, network surface and precision

#### M.25.2.2 Section 2

**Section title: Web Security**

**Topics covered in this section:**

- Injection Flows
- Cookies Flows
- Server Misconfiguration

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

- What is the difference between boolean-based and time-based SQL injection?
- Can regex matching protect against Directory Traversal attack?
- Does the Same Origin Policy apply to the localStorage inside the browser?

**Typical questions for seminar classes (labs) within this section**

- Vulnerability analysis and exploitation for a given web application
- Write and deploy WAF rules to mitigate a specific web attack

**Test questions for final assessment in this section**

As above

#### M.25.2.3 Section 3

**Section title: Software Security**

**Topics covered in this section:**

- Buffer overflow
- Format string
- ASLR
- NX
- Fuzzing

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

- What are the pros and cons of using ASLR? does it affect the performance?
- What is the required information to be able to identify a remote libc version?
- What are the pros and cons of writing your own fuzzer?

**Typical questions for seminar classes (labs) within this section**

- Write an exploit for a given binary, also try to bypass the mitigation techniques
- Implement a fuzzer for a specific use-case

**Test questions for final assessment in this section**

As above

#### M.25.2.4 Section 4

**Section title: GSM**

**Topics covered in this section:**

- Um interface
- IMSI/TMSI
- A5/1 stream cipher
- SIM & USIM cards, A3/A8 and COMP128

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

- What are 2G, UMTS, LTE, LTE-A, ARFCNs, and hopping channels?
- What is a fake or rogue Base Transceiver Station?
- What frequency ranges are we listening to?
- What frequency bandwidth is required
- What SDR devices are compatible with those requirements?
- What is an IMSI versus a TMSI and an MSISDN?

**Typical questions for seminar classes (labs) within this section**

- How to differentiate 2G vs UMTS vs LTE channels?
- What telecom operators do you identify on 2G GSM900 and DCS1800?
- What telecom operators do you identify on LTE and on what bands?
- Locate cell towers and attempt to locate subscribers (yourself)
- Trace your own IMSI and eavesdrop on a circumvented 2G SMS and voice call
- Deal with hopping either by capturing a larger bandwidth, or using two RTL dongles with the same phase

**Test questions for final assessment in this section**

- Briefly describe a known attack against the A5/1 stream cipher
- How does COMP128 work? Is it vulnerable?
- Name various ways to deal with 2G channel hopping
- How often do IMSIs show up and in what situations does that usually happen?

## M.26 Optimization

- **Course name:** Optimization
- **Course number:** XYZ
- **Knowledge area:** Data Science

### M.26.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 3rd year of BS
- **Semester of instruction:** 2nd semester
- **No. of Credits:** 5 ECTS
- **Total workload on average:** 180 hours overall
- **Class lecture hours:** 2 per week.
- **Class tutorial hours:** 2 per week.
- **Lab hours:** 2 per week.
- **Individual lab hours:** 0.
- **Frequency:** weekly throughout the semester.
- **Grading mode:** letters: A, B, C, D.

### M.26.2 Prerequisites

Familiarity with multivariate calculus, elementary real analysis, and linear algebra.

- Multivariate calculus
- Linear Algebra
- Probability and Statistics

### M.26.3 Course outline

This course studies fundamental concepts of optimization, which is the problem of making decisions to maximize or minimize an objective in the presence of complicating constraints. The course takes a unified view of optimization and covers the main areas of application and the main optimization algorithms. It covers the following main topics: Linear optimization, Network flows, Discrete optimization, Dynamic optimization, and Nonlinear optimization.

### M.26.4 Expected learning outcomes

After this course, students will be able to:

- be able to formulate problems in their fields of research as optimization problems
- be able to transform an optimization problem into its standard form
- understand how to assess and check the feasiblity and optimality of a particular solution to a general constrained optimization problem
- be able to use the optimality conditions to search for a local or global solution from a starting point.
- understand the computational details behind the numerical methods discussed in class, when they apply, and what their convergence rates are.
- be able to implement the numerical methods discussed in class and verify their theoretical properties in practice.
- be able to apply the learned techniques and analysis tools to problems arising in their own research.

### M.26.5 Expected acquired core competences

- Numerical analysis
- Numerical optimization

### M.26.6 Detailed topics covered in the course

- Linear Optimization
- Simplex Methods
- Duality Theory
- Sensitivity Analysis
- Network Flows
- Branch and Bound and Cutting planes

- Lagrangean Methods
- Dynamic Programming
- Nonlinear Optimization
- Optimality conditions and Gradient methods
- Markov Chains and Markov Decision Processes
- Stochastic Optimization

### M.26.7 Textbook

- Bertsimas, Dimitris, and John Tsitsiklis. *Introduction to Linear Optimization*
- J. Nocedal and S. Wright. *Numerical Optimization*

### M.26.8 Reference material

NA

### M.26.9 Required computer resources

NA

### M.26.10 Evaluation

- Quizzes (20%)
- Labs (10%)
- Midterm Exam (25%)
- Final Exam (25%)
- Class and lab participation (10%)

## M.27 Personal Software Process

- **Course name:** Personal Software Process
- **Course number:** F19-SE-PSP
- **Area of instruction:** Computer Science and Engineering

### M.27.1 Course Characteristics

#### M.27.1.1 Key concepts of the class

A Practitioner's Start-Up Kit course introduces the highest-leverage metrics, specifically the ones associated with improving time estimation and reducing defects. PSP is intended for practicing software engineers and their managers. The measures introduced can serve as the basis for software development process improvement in the organization as well as helping individuals.

#### M.27.1.2 What is the purpose of this course?

The main objectives of this course are the following:

- Student will reduce overall defect rates
- Student will spend more time at the front end of the development cycle
- Student will eliminate or nearly eliminate compile and test defects
- Student will be able to more accurately estimate the time it takes to build software

#### M.27.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to ...

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to ...

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

### M.27.1.4 Course evaluation

**Table M.69:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Assignments | 50 | 50 |
| Reports | 40 | 40 |
| Attendance | 10 | 10 |

**Guidelines**

Instructor reviews your homework assignments for the following elements:

- Check if the data presented is within "reasonable" bounds for the current assignment. Reasonable bounds is dependent upon each case depending upon the assignment and the tools being used by you to develop the assignment.
- Identify any spikes or unusual patterns in the data which may be an indication of incorrect application of the process, a simple typo or incorrectly stated assumptions.
- Collate data submitted by multiple students in order to come up with a common set of numbers reflecting the data of the entire class.

Assignments will be marked as either, PASS, PASS but, or Resubmit.

- PASS - Means everything is good and you can proceed. Scored as a 0 (zero).
- PASS but - Means you did pretty well, but there are some minor things to review and in the long run these might create problems. Scored as a 0.5 (zero point five).
- Resubmit - You made an error that will cause many issues later on so it must be corrected before your proceed. Scored as a 1 (one).

Objective of course is to complete all assignments with the minimum score possible. i.e. a total score of zero at the end means you got a perfect score.

A passing grade indicates that you have successfully demonstrated an understanding of the lecture material.

#### M.27.1.5 Grades range

**Table M.70:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| P. Pass | 80-100 | 85-100 |
| B. Fail | 0-80 | 75-84 |

#### M.27.1.6 Resources and reference material

- Humphrey, Watts S. PSP: A Self-Improvement Process for Software Engineers. Reading, MA: Addison Wesley, 2005. (Prefered)
- Humphrey, Watts S. Introduction to the Personal Software Process. Reading, MA: Addison Wesley, 1997.
- Humphrey, Watts S. Discipline for Software Engineering: The Complete PSP Book. Reading, MA: Addison Wesley, 1995. (Contains original exercises)

### M.27.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.71:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction | 4 |
| 2 | Planing and Estimation | 6 |
| 3 | Quality management | 6 |
| 4 | Time management | 4 |
| 5 | Conclusion | 4 |

#### M.27.2.1 Section 1

**Section title:** Introduction

**Topics covered in this section:**

- Introduction,
- The Dashboard Tool

- Collecting Data

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical tasks and exercises for ongoing performance evaluation within this section**

1. Reading (Hum 05): Chapter 1 2
2. Skim data collection reading

**Typical tasks and exercises for seminar classes (labs) within this section**

1. Install the Dashboard Tool
2. Write Program using PSP0.0

**Test questions for final assessment in this section**

1. ...

#### M.27.2.2 Section 3

**Section title:** Planing and Estimation

**Topics covered in this section:**

- The Planning Process
- Software Size Estimation Methods
- The PROBE Size Estimation Method

### What forms of evaluation were used to test students' performance in this section?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

### Typical tasks and exercises for ongoing performance evaluation within this section

1. ...

### Typical tasks and exercises for seminar classes (labs) within this section

1. ...

### Test questions for final assessment in this section

1. ...

### M.27.2.3 Section 3

**Section title:** Quality management

**Topics covered in this section:**

- Design Code Reviews
- Software Quality Management
- Process Definition
- Resource Schedule Estimation

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical tasks and exercises for ongoing performance evaluation within this section**

1. ...

**Typical tasks and exercises for seminar classes (labs) within this section**

1. ...

**Test questions for final assessment in this section**

1. ...

### M.27.2.4 Section 3

**Section title:** Time management

**Topics covered in this section:**

- Time Management
- Pomodoro - Concentration

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical tasks and exercises for ongoing performance evaluation within this section**

1. ...

**Typical tasks and exercises for seminar classes (labs) within this section**

1. ...

**Test questions for final assessment in this section**

1. ...

#### M.27.2.5 Section 5

**Section title:** Conclusion

**Topics covered in this section:**

- The PSP Body of Knowledge (BoK)
- Beyond PSP  TSP

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical tasks and exercises for ongoing performance evaluation within this section**

1. ...

**Typical tasks and exercises for seminar classes (labs) within this section**

1. ...

**Test questions for final assessment in this section**

1. ...

## M.28 Requirements Engineering

- **Course name:** Requirements Engineering
- **Course number:** SE-

### M.28.1 Course characteristics

#### M.28.1.1 Key concepts of the class

- Requirements elicitation
- Requirements specification
- Requirements prototyping and implementation
- Requirements verification
- Requirements traceability

#### M.28.1.2 What is the purpose of this course?

The course has the following key objectives:

- To introduce the motivation, conceptual background and terminology on which requirements engineering relies.
- To provide a comprehensive account of state-of-the-art techniques for requirements engineering.
- To let the students experience the actual requirements-caused problems faced by real software teams.

#### M.28.1.3 Course Objectives Based on Bloom's Taxonomy

The "Requirements Engineering" course develops students' skills at all the 6 levels of the Bloom's taxonomy.

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define:

- System requirements
- Software requirements

- Domain knowledge
- Environment assumptions
- Environment-controlled phenomena
- Machine-controlled phenomena
- Environment-observed phenomena
- Machine-observed phenomena
- Problem space
- Solution space
- Prescriptive statements
- Descriptive statements
- Traceability links

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain (with examples):

- Difference between system and software requirements
- Difference between domain knowledge and environment assumptions
- Pairwise difference between environment- and machine-controlled (observed) phenomena
- Difference between the world and the machine
- Difference between problem and solution space
- Difference between prescriptive and descriptive statements
- Difference between vertical and horizontal traceability

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply:

- Requirements elicitation techniques
- Requirements specification techniques
- Prototyping and implementation techniques
- Negotiation techniques for modifying requirements
- Techniques for establishing traceability links, both vertical and horizontal

- Parameterized unit testing
- Acceptance testing

**- What inference can a student make based on the acquired knowledge?**

By the end of the course, the students should be able to identify:

- Lack of traceability links
- Incorrectly implemented requirements
- Incorrectly elicited requirements
- Incompletely implemented requirements
- Incompletely elicited requirements

**- What judgements can a student make about the studied field?**

By the end of the course, the students should be able to judge:

- Completeness of a requirements document specified by others
- Correctness of a requirements document specified by others
- Completeness of an implementation developed by others wrt requirements
- Correctness of an implementation developed by others wrt requirements
- Traceability for software artifacts created by others
- Presentations of other students

**- What actions can students take based on their judgement?**

By the end of the course, the students should be able to take appropriate actions for:

- Eliciting lacking requirements from stakeholders
- Negotiating requirements modifications with stakeholders
- Implementing lacking functionality, wrt to requirements, in software developed by others
- Fixing functionality that incorrectly implements requirements in software developed by others
- Introducing missing traceability links
- Writing additional tests to achieve sufficient requirements coverage

#### M.28.1.4 Course evaluation

### M.28.2 Evaluation

**Table M.72:** Course grade breakdown

| Type | Default points |
|---|---|
| Practical assignments | 60 |
| Reading assignments | 18 |
| Project presentations | 12 |
| Classroom participation | 10 |

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

#### M.28.2.1 Grades range

**Table M.73:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 65-79 |
| C. Satisfactory | 60-74 | 50-64 |
| D. Poor | 0-59 | 0-49 |

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table M.73, but it may change slightly (usually reduced) depending on how the semester progresses.

#### M.28.2.2 Resources and reference material

- Handouts supplied by the instructor
-
-
-

## M.28.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.74:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Requirements elicitation and documentation | 20 |
| 2 | Requirements prototyping and implementation | 20 |
| 3 | Requirements verification and traceability | 20 |

### M.28.3.1 Section 1

**Section title:**

Requirements elicitation and documentation

**Topics covered in this section:**

- Foundations of requirements engineering
- The world and the machine
- Domain understanding and requirements elicitation
- Questions for interviews
- The requirements process
- Use cases
- Requirements specification and documentation

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the WHY-Dimension of requirements engineering?
2. What criteria are recommended to use for stakeholders analysis?
3. Who is a stakeholder?
4. What is an artifact-driven elicitation technique?
5. What are the four principles for description in requirements engineering?
6. What are the four facets of relationship between the world and the machine?
7. What are the four kinds of denial in software engineering?
8. What is a descriptive statement?
9. What are the different kinds of information about the world?

**Typical questions for seminar classes (labs) within this section**

1. Write down and present a project proposal for implementing during the course.
2. Propose a set of questions for a requirements elicitation interview.
3. Conduct, audio record and transcribe an elicitation interview.
4. Design use cases based on the elicitation transcript and audio recording.

**Test questions for final assessment in this section**

1. Present you experience of preparing and conducting the elicitation interview.
2. How did you choose the stakeholder for interviewing?
3. Did the interview go according to the plan?
4. Which of the initially prepared questions you did not ask during the interview? Why?
5. What questions you had to ask in addition to the initially prepared ones? Why?
6. If you have been interviewed, how relevant were the interviewer's questions?
7. What conflicts did you have when merging the interview transcripts of your team members?
8. How did you solve the merging conflicts?
9. What lessons have you learned based on your experience as an interviewer and an interviewee?
10. Present use cases constructed based on the elicited information.
11. How do the use cases trace to the interview transcript?

12. How does the interview transcript trace to the use cases?

#### M.28.3.2 Section 2

**Section title:**

Requirements prototyping and implementation

**Topics covered in this section:**

- Mapping use cases to object models
- From use cases to user interface design
- Activity diagrams
- The psychopathology of everyday things
- Seamless requirements
- The anatomy of requirements

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What value do UML diagrams bring to the requirements engineering process?
2. Define the "extend" relationship between use cases.
3. Enumerate risk reduction tactics.
4. Describe defining characteristics of what Davis calls "knowledge structure".
5. What activities does the risk management process involve?
6. How do you call an active component in a use case diagram?
7. What is the purpose of postconditions in use cases?
8. Name different types of relationships in use case modelling.
9. Categorize UML as either informal, semi-formal or formal notation.
10. Define the "generalization" relationship in UML.

**Typical questions for seminar classes (labs) within this section**

1. Reflect, individually and in teams, on the use cases and the use case diagram that you received for implementation from another team.
2. Construct activity diagrams from the use cases and the use case diagram.
3. Construct classes based on the activity diagrams.
4. Design user interfaces based on the classes and activity diagrams.
5. Develop a minimum viable product (MVP) implementing your input requirements.

**Test questions for final assessment in this section**

1. Record and present a short demo of your MVP.
2. What decisions did you have to take when implementing the MVP?
3. How did you define your MVP?
4. What did you have to change in the requirements document, and why?
5. What requirements you decided to cover and not to cover in the MVP, and why?
6. Present lessons learned from developing the MVP.

### M.28.3.3 Section 3

**Section title:**

Requirements verification and traceability

**Topics covered in this section:**

- Parameterized unit tests
- Goal modelling
- Scrum & User stories
- Use case testing

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. How do we call an active system component playing a specific role in goal satisfaction?
2. How do we call an autonomous and passive object in the object model, which cannot control the behaviours of instances of other objects?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 1 |
| Oral polls | 1 |
| Discussions | 1 |

3. What is a goal?
4. What goal pattern refers to every future state?
5. How do we call an association where the composite object and its components appear and disappear together in the system?
6. Describe the goal refinement process.
7. What is object specialization?
8. Define the “maintain” goal pattern.
9. Enumerate and describe different relationships between goals.

**Typical questions for seminar classes (labs) within this section**

1. Construct parameterized unit tests for the MVP provided to you by another team.
2. Reflect, individually and in teams, on the MVP provided to you by another team.
3. Reflect, individually and in teams, on the user interface design of the MVP.
4. Develop the MVP into a usable production-quality software.
5. Construct use case tests and parameterized units tests for the final implementation; update the requirements document as you go.
6. Run the tests and fix the identified defects; update the requirements document as you go.
7. Ensure pairwise mutual completeness between the requirements, final implementation and tests.
8. Ensure pairwise mutual traceability between the requirements, final implementation and tests.
9. Write a document that will describe very clearly how to run and use the final implementation.
10. Describe in the document how to reproduce different use cases in the actual

software.

**Test questions for final assessment in this section**

1. Present the final system developed from the MVP received from another team.
2. Introduce the project and its business goals.
3. Evaluate the quality of the interview transcript.
4. Evaluate the quality of the use cases.
5. Evaluate the quality of the MVP and user interfaces.
6. Reflect on the quality management process.
7. Record and present demo of test runs.
8. Reflect on teamwork and communication with other teams.
9. Present lessons learned while implementing different parts of different projects coming from the other teams.
10. Record and present demo runs of the software using the use cases as the reference.
11. Describe strengths and weaknesses of the final implementation.
12. Write an essay detailing your reflections on the overall course experience.

# M.29 Security of Systems and Networks

- **Course name:** Security of Systems and Networks (SSN)
- **Course number:** SNE-08

## M.29.1 Course characteristics

### M.29.1.1 Key concepts of the class

- Network security
- Applied cryptography
- Security protocols
- Internet security

### M.29.1.2 What is the purpose of this course?

This course will cover the fundamentals of security, security protocols, and their applications in real-world. The topics covered in this course include crypto, authentication, passwords, practical security, social aspects of security, SSL/TLS, email security, PKI, and IPSec. Furthermore, this course will strengthen the security knowledge of the students and guide them in the right direction for their upcoming research projects. The course is divided into two parts. The first part will cover the theory and hands-on practice of the concepts taught at class. And the second part of the course will focus on the course projects. The student will work on a security project by using the concepts taught in the class.

### M.29.1.3 Course Objectives Based on Bloom's Taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to recognize and define

- Different aspects of system and network security
- Authentication protocols
- Various key management protocols
- Different symmetric and asymmetric ciphers

- Security at different TCP/IP layers
- Nuts and bolts of quantum cryptography

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to:

- Demonstrate the acquired knowledge and skills in applied cryptography (symmetric and asymmetric cryptography),
- Operate classical enigma machine, encode and decode messages with it
- Demonstrate the working knowledge of famous cryptographic algorithms and discuss their shortcomings
- Demonstrate and operate the already implemented security protocols over internet,
- Reason about the problems in the security of networked systems and current internet and their existing solutions,
- Solve mathematical problems (especially in number theory),
- And Demonstrate the knowledge and discuss basic quantum cryptography concepts.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to apply

- Crypt-analyze ciphertext and decrypt through frequency analysis and other important techniques
- Design security protocols
- Find security flaws in security protocols
- Get hands-on experience of the existing enterprise cryptographic algorithms and use them in projects,
- Demonstrate the skill of finding out security issues in networked systems and internet technologies,

**Table M.75:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 35 | 35 |
| Project | 35 | 35 |
| Exams | 30 | 30 |

#### M.29.1.4 Course evaluation

If necessary, please indicate freely your course's features in terms of students' performance assessment: None

#### M.29.1.5 Grades range

**Table M.76:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

If necessary, please indicate freely your course's grading features: The semester starts with the default range as proposed in the Table M.76, but it may change slightly (usually reduced) depending on how the semester progresses.

#### M.29.1.6 Resources and reference material

- Lecture material
- RFCs
- Information Security by Mark Stamp

### M.29.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table M.77:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Classical and modern cryptography | 8 |
| 2 | Authentication | 8 |
| 3 | SSL, TLS, and IPSec | 10 |
| 4 | Quantum cryptography | 2 |
| 5 | Labs | 56 |

#### M.29.2.1 Section 1

**Section title:**

Classical and modern cryptography

**Topics covered in this section:**

- Classical cryptography
- Enigma
- Different substitution and transposition ciphers
- Stream and block ciphers
- Data Encryption Standard (DES)
- Advanced Encryption Standard (AES)
- Diffie-Hellman key exchange
- Crypto math
- RSA
- Elliptic curve cryptography

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. How do different protocols work?
2. What are the differences in stream and block ciphers from performance standpoint?

3. How to measure the security of cryptographic algorithms?
4. How to encrypt and decrypt with different asymmetric crypto algorithms?
5. How to embedd backdoors in crypto algorithms?
6. How to realize key exchange through diffie-hellman using traditional techniques and elliptic curve techniques?
7. How to make security algorithms efficient?

**Typical questions for seminar classes (labs) within this section**

1. Make enigma machine with pringle box
2. Assess the security of different setups of RSA
3. Implement man in the middle attack
4. Implement addition over elliptic curves
5. Solve crypto math problems

**Test questions for final assessment in this section**

1. As above

#### M.29.2.2 Section 2

**Section title:**

Authentication

**Topics covered in this section:**

- Kerberos
- Passwords
- Biometrics
- Authentication and key agreement protocols
- Protocol development

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. What are pros and cons of using symmetric and asymmetric cryptographic mechanisms for authentication?
2. What is man in the middle attack?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

3. Develop home-grown authentication mechanisms?
4. How Kerberos reduces the communication overhead?
5. Where is shibboleth used?

**Typical questions for seminar classes (labs) within this section**

1. Implement different variants of authentication protocols
2. Find out security flaws in authentication protocols
3. Identify shortcomings of different protocols

**Test questions for final assessment in this section**

1. As above

#### M.29.2.3 Section 3

**Section title:**

SSL, TLS, and IPSec

**Topics covered in this section:**

- SSL, TLS
- IPSec

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. How does SSL and TLS work?
2. HOw does SSL and TLS combine symmetric and asymmetric cryptography?
3. Why IPSec is so over-engineered? and what are the security flaws?
4. What are different components of IPSec

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

### Typical questions for seminar classes (labs) within this section

1. Implement IPSec
2. Assess the security of SSL and TLS handshakes

### Test questions for final assessment in this section

1. As above

### M.29.2.4 Section 4

### Section title:

Quantum cryptography

### Topics covered in this section:

- Superposition of photons
- Quantum Key Distribution (QKD)
- Eavesdropper in quantum setup

### What forms of evaluation were used to test students' performance in this section?

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

### Typical questions for ongoing performance evaluation within this section

1. How is eavesdropper kept at bay from eavesdropping on a quantum channel?

2. What are the advantages of quantum channel over traditional communication channel?

**Test questions for final assessment in this section**

1. As above

## M.30 Sensing, Perception & Actuation

- **Course name:** Sensing, Perception & Actuation
- **Course number:** R-05
- **Area of instruction:** Computer Science and Engineering

### M.30.1 Administrative details

- **Faculty:** Computer Science and Engineering
- **Year of instruction:** 1st year of MSc
- **Semester of instruction:** 1st semester
- **No. of Credits:** 5 ECTS
- **Total workload on average:** 180 hours overall
- **Frontal lecture hours:** 2 hours per week.
- **Frontal tutorial hours:** 0 hours per week.
- **Lab hours:** 2 hours per week.
- **Individual lab hours:** 2 hours per week.
- **Frequency:** weekly throughout the semester.
- **Grading mode:** letters: A, B, C, D.

### M.30.2 Course outline

This course covers selected topics in sensors and sensing area, which are in particular important for robotic application. The students are expected to learn the course topics on their own beyond the level of the lectures. The goals throughout the course are to refresh students' math skills in data and error analysis, and familiarize them with sensing principles and sensor utilization, giving them both analytic and experimental experience. The students will be required to participate in laboratory practicum and solve practical tasks on data processing in MATLAB environment.

### M.30.3 Expected learning outcomes

- Be acquainted with and utilize main sensors for robotic applications.

- Be familiar with Sensing principles, Measurements and error analysis, Data analysis, Sensor calibration, etc.
- Be able to write a code for sensor data processing in MATLAB.
- Train engineering skills during lab. tests, studying sensor calibration and utilization, and using relevant software for sensor data capture, export, interpretation and representation.

### M.30.4 Required background knowledge

- XXX

### M.30.5 Prerequisite courses

- Physics I
- Physics II
- Mathematical Analysis I
- Mathematical Analysis II
- Analytic Geometry and Linear Algebra I
- Analytic Geometry and Linear Algebra II
- Probability and Statistics

### M.30.6 Detailed topics covered in the course

The topics below are presented with the granularity of at most the academic hour of instruction. For each topic it is specified if it an **I**ntroduction to the topic, a **D**eep explanation, or a **R**eview of a subject already covered in another course.

- Introduction to sensors: Sensors' classification, Characteristics, Dynamic range Accuracy
- Introduction to Measurements and Error Analysis
- Introduction to Data Analysis: Linear Regression, Least-Squares Fitting, Curve fitting, and Filtering
- Image sensors: camera matrix, characteristics and calibration
- Video camera: CCTV, IR & thermal imaging camera, Fish eye camera
- Stereo vision: Stereosystem, Stereogeometry, 3D reconstruction

- Depth, TOF, RGBD camera, MS Kinect: characteristics and calibration
- Review and Midterm
- Sensor fusion (principles), Multisensory, multicamera, MoCap systems
- LIDAR: Laser rangefinders. Laser-camera systems
- SONAR. Doppler radar. Acoustic sensor systems. Sound spectrogram
- Inertial sensors: IMU, accelerometers, gyroscopes, Magnetic Compasses, GPS
- Internal sensors: position, velocity, torque & force sensors, encoders
- MEMS for robot applications. Smart and Intelligent Sensors

### M.30.7 Textbook

-

### M.30.8 Reference material

- Slides will be provided during the course

### M.30.9 Required computer resources

Students are required to have laptops.

### M.30.10 Evaluation

- Home assignments (25%)
- Mid-term exam (15%)
- In-class activity, quizzes and lab. practicum (15%)
- Final exam (25%)
- Project (20%)

# Appendix N

# Catalogue of the elective courses

The elective courses administered in the programs are:

1. Information Retrieval for Data Science (see Section N.1 on page 849)
2. ...

They are divided across programs as follows:

- **Software Engineering**
  1. ...
- **Robotics**
  1. ...
- **Data Science**
  1. Information Retrieval for Data Science (see Section N.1 on page 849)
  2. ...
- **Secure Systems and Network Engineering**
  1. ....

## N.1 Information Retrieval for Data Science

- **Course name:** Information Retrieval for Data Science
- **Course number:** N/A

### N.1.1 Course Characteristics

### N.1.2 What subject area does your course (discipline) belong to?

Computer systems organization; Information systems; Real-time systems; Information retrieval; World Wide Web

#### N.1.2.1 Key concepts of the class

- Data indexing
- Relevance and ranking

#### N.1.2.2 What is the purpose of this course?

The course is designed to prepare students to understand and learn contemporary tools of information retrieval systems. Students, who will later dedicate their engineering or scientific careers to implementation of search engines, social networks, recommender systems and other content services will obtain necessary knowledge and skills in designing and implementing essential parts of such systems.

#### N.1.2.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and recognize

- Terms and definitions used in area of information retrieval,
- Search engine and recommender system essential parts,
- Quality metrics of information retrieval systems,
- Contemporary approaches to semantic data analysis,
- Indexing strategies.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to describe and explain

- How to design a recommender system from scratch,
- How to evaluate quality of a particular information retrieval system,
- Core ideas and system implementation and maintenance,
- How to identify and fix information retrieval system problems.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to

- Build a recommender service from scratch,
- Implement proper index for an unstructured dataset,
- Plan quality measures for a new recommender service,
- Run initial data analysis and problem evaluation for a business task, related to information retrieval.

#### N.1.2.4 Course evaluation

**Table N.1:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 0 |
| Interim performance assessment | 30 | 0 |
| Assessments | 0 | 50 |
| Exams | 50 | 50 |

If necessary, please indicate freely your course's features in terms of students' performance assessment:

Labs are followed by the home works. Home works are covering 50% of the grade. 50% of the grade fall to exam session, which will be in the form of project defence.

#### N.1.2.5 Grades range

#### N.1.2.6 Resources and reference material

Main textbook:

**Table N.2:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 60-79 |
| C. Satisfactory | 60-74 | 40-59 |
| D. Poor | 0-59 | 0-39 |

- "An Introduction to Information Retrieval" by Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Cambridge University Press (any edition)

Other reference material:

- "Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit", Steven Bird, Ewan Klein, and Edward Loper. [link]

## N.1.3 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

**Table N.3:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Text processing (1, 3, 4, 7, 8, 9) | 24 |
| 2 | Text indexing and search (2, 5) | 8 |
| 3 | Web basics and algorithms for web (6, 10, 14) | 12 |
| 4 | Media types processing: images, video, sound (11, 12) | 8 |
| 5 | Quality assessment (13, 15) | 8 |

### N.1.3.1 Section 1

**Section title:** Text processing

**Topics covered in this section:**

- Introduction to information retrieval
- Distributional semantics. Vector model. Dimension reduction
- ML approaches to vector modelling

- Spellchecking and query correction
- Query expansion and suggest
- Language model. Topic model. Clustering and classification

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Build term-document matrix from the text.
2. Visualize a dataset in 2D using PCA.
3. Find K most similar text using ANN-based language model (embeddings).
4. Implement Levenstein distance function.
5. Discover 5 major topics in a text dataset.

**Typical questions for seminar classes (labs) within this section**

1. Build term-document matrix of a set of web pages using TF-IDF.
2. Implement PCA-based latent semantic analysis for a set of text with labels. Visualize classes (labels) in 2D.
3. Train Doc2vec model and implement search with ranking using cosine similarity.
4. Implement spellchecking algorithm.
5. Discover optimal number of topics in a dataset using topic modelling.

**Test questions for final assessment in this section**

1. What are the approaches to vector space modelling for text datasets?
2. What are the most common techniques to process text before indexing?
3. Explain topic modelling problem statement.

#### N.1.3.2 Section 2

**Section title:** Text indexing and search

**Topics covered in this section:**

- Building inverted index.
- Language, tokenization, stemming, searching, scoring.
- Indexing for vector model.
- Kd-trees, Quad-trees, Annoy, FAISS, HNSW index.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Build inverted index for a text.
2. Tokenize a text.
3. Write kd-tree.
4. Implement small-world graph.
5. Build HNSW index.

**Typical questions for seminar classes (labs) within this section**

1. Build inverted index for a set of web pages.
2. build a distribuition of stems/lexemes for a text.
3. Choose and implement persistent index for a given text collection.
4. Compare 2 logarithmic index structures for a given task.
5. Choose and implement updateable index for vector space of a given dimensions.

**Test questions for final assessment in this section**

1. Explain how (and why) KD-trees work.

2. What are weak places of inverted index?
3. State the problem of approximate nearest neighbours search. Overview solutions.
4. Discuss HNSW index architecture.

#### N.1.3.3 Section 3

**Section title:** Web basics and algorithms for web

**Topics covered in this section:**

- Web basics. Internet crawling. XML and HTML processing. Dynamic documents processing.
- On newsfeeds.
- Web search specific topics. PageRank. Duplicates. CTR.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Extract text from web page.
2. Extract text from dynamic web page.
3. Crawl a website with respect of robots.txt.
4. Explain PageRank.

**Typical questions for seminar classes (labs) within this section**

1. Write a crawler for a dynamically generated web site.
2. Build a graph of a website.
3. Run a PageRank on a graph of a single website.
4. Compute CTR for a given log and train a relevance model using this data.

**Test questions for final assessment in this section**

1. Explain what is DOM (document object model) and how is it used in document parsing?
2. What is the difference in crawling static and dynamic pages?
3. What are the weaknesses of PageRank?

### N.1.3.4 Section 4

**Section title:** Media types processing: images, video, sound

**Topics covered in this section:**

- Image and video processing.
- Image understanding
- Image enhancing
- Video understanding
- Audio processing
- Speech-to-text

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Extract 4 major colors from image.
2. Split video into shots.
3. Identify a chord in audio.
4. Text-2-speech.
5. Label objects in images.

**Typical questions for seminar classes (labs) within this section**

1. Build a "search by color" feature.
2. Extract scenes from video.
3. Build a voice fingerprint and identify a person by voice.
4. Write a voice-controlled search.
5. Semantic search withing unlabelled image dataset.

**Test questions for final assessment in this section**

1. What the the approached to image understanding?
2. How to cluster a video into scenes and shots?
3. How speech-to-text technology works?
4. How to build audio fingerprints?

#### N.1.3.5 Section 5

**Section title:** Quality assessment

**Topics covered in this section:**

- Search quality assessment.
- Recommender quality assessment.
- A/B testing.
- SBS.
- pFound, DCG, nDCG.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 1 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 0 |

**Typical questions for ongoing performance evaluation within this section**

1. Propose a strategy for A/B testing.
2. Propose recommender quality metric.
3. Implement DCG metric.
4. Discuss relevance metric.

**Typical questions for seminar classes (labs) within this section**

1. Provide a framework to accept/reject A/B testing results.
2. Compute DCG for an example query for random search engine.
3. Implement a metric for a recommender system.
4. Implement pFound.

**Test questions for final assessment in this section**

1. What is SBS (side-by-side) and how is it used in search engines?
2. Compare pFound with CTR and with DCG.
3. Explain how A/B testing works.

## N.2 Fundamentals of Robot Control

- **Course name:** Fundamentals of Robot Control
- **Course number:** F19

### N.2.1 Course Characteristics

#### N.2.1.1 Key concepts of the class

- Introductory nonlinear control over dynamic systems with the focus on robotics
- Stability, pros and cons of nonlinear control systems

#### N.2.1.2 What is the purpose of this course?

Control theory is an integral part of modern robotics, and there is a high chance that most students majoring in Robotics would face the problems of controlling a physical plant (a robot, manipulator, drone, autonomous vehicle) in their research and graduation work as well as their professional careers. Therefore, the main purpose of this course is to prepare the students for solving practical control problems by teaching the most fundamental approaches of nonlinear control used in modern robotics applications.

#### N.2.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to remember and differentiate

- Basic structure of differential equations describing motion of robotic manipulators,
- Motivation behind and the basic structure of feedback control systems,
- How to find control system's error dynamics and methods to analyze it,
- General structure of linear controllers (P, PD, PID),
- Physical motivation behind Lyapunov stability analysis,
- Basic structure of robust control system.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to

- Name the main sources of nonlinearities in physical systems,
- Explain the cons of applying PID controllers to nonlinear systems,
- Understand pros and cons of feedback linearization method,
- Name pros and cons of robust control approach,
- Numerically solve differential equations in MATLAB environment.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

- Know how to analyze stability of physical systems with Lyapunov direct method,
- Design feedback linearization systems,
- Synthesize robust control systems and tune them,
- Implement nonlinear control in MATLAB environment to simulate the behavior of multi-DOF robotic systems.

#### N.2.1.4 Course evaluation

**Table N.4:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | 20 | 0 |
| Interim performance assessment | 30 | 60 |
| Exams | 50 | 40 |

If necessary, please indicate freely your course's features in terms of students' performance assessment:

The course grades are given according to the following rules: Homework assignments (4) = 20 pts, Quizzes (4) = 40 pts, Term project = 40 pts.

**Table N.5:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | 80-100 |
| B. Good | 75-89 | 60-79 |
| C. Satisfactory | 60-74 | 40-59 |
| D. Poor | 0-59 | 0-39 |

#### N.2.1.5 Grades range

If necessary, please indicate freely your course's grading features: The first year master course students come with very diverse backgrounds, and therefore the grade requirements for this course are relaxed.

#### N.2.1.6 Resources and reference material

The course is build based on these main textbooks:

- "Applied nonlinear control," J.-J. Slotine & Weiping Li. Pearson, 1991.
- "Robotics: modelling, planning and control," Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. Springer Science & Business Media, 2010.

Other reference material:

- "Robot modeling and control," Mark Spong, Seth Hutchinson, M. Vidyasagar. John Wiley & Sons, 2006.
- "Modern Control Systems" (13th Edition), Richard Dorf & Robert H. Bishop. Pearson, 2016.
- "Modern Robotics: Mechanics, Planning, and Control," Kevin Lynch, Frank Park. Cambridge University Press, 2017 (*also, check out their video materials on YouTube*).

### N.2.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

Table N.6: Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Motion. Kinematics. Dynamics | 7 |
| 2 | Linear systems. Stability | 6 |
| 3 | Feedback control systems | 3 |
| 4 | Feedback linearization | 6 |
| 5 | Robust control | 8 |

#### N.2.2.1 Section 1

**Section title:** Motion. Kinematics. Dynamics.

**Topics covered in this section:**

- Free body motion
- Manipulator position and orientation
- Homogeneous transformations
- Forward and inverse kinematics
- Kinetic and potential energy
- Euler-Lagrange equations

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Given initial and final object position and orientations, obtain the corresponding transformation matrix.
2. Find the Jacobian for a given manipulator.
3. For a given differential equation describing a physical system, and for given kinetic (K) and potential energies (U) do the following:
    - Show that there always exists a solution with respect to acceleration and that it is unique;

- Demonstrate that the system's Hamiltonian H = K + U remains constant in the absence of external torques and forces;
- Show that the rate of change of the total energy equals instantaneous mechanical power.

4. For a given manipulator with known Jacobian, do the following:
   - Find kinetic and potential energies of the robot;
   - Drive the Euler-Lagrange differential equation of motion.

**Typical questions for seminar classes (labs) within this section**

1. Do the following with MATLAB Robotics Toolbox:
   - Compute basic rotation and homogeneous transformation matrices;
   - Create a two-link manipulator and solve its forward kinematics;
   - Simulate forward dynamics;
   - Compute inertial, Coriolis and gravitational forces for the manipulator.
2. Compute and analyze inertial tensor for a given robotic manipulator;
3. For manipulators with different kinematic configurations, analytically derive their Jacobian matrices;
4. Derive and analyze Euler-Lagrange equations describing dynamics of a given manipulator (required torques, singularities).

**Test questions for final assessment in this section**

1. What is a rotation matrix and what does it describe?
2. How to find a homogeneous transformation matrix? How is it different from rotation matrix?
3. What is manipulator Jacobian? How does it relate static forces and torques? How can one use the Jacobian to analyze manipulator singularities?
4. What is physical nature of the terms of Euler-Lagrange equations of robot motion?
5. What are the main properties of the basic terms of differential equations of motion (invertibility, positive definiteness, singularities, limits).

#### N.2.2.2 Section 2

**Section title:** Linear systems. Stability

**Topics covered in this section:**

- State-space equations
- Eigenvalues and eigenvectors
- Phase plane analysis
- Energy and stability
- Lyapunov's direct method
- Lyapunov stability analysis

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Convert given differential equation into state-space form
2. Find eigenvalue for a given matrix
3. Show that given a constant matrix $\mathbf{M}$ and any time-varying vector $\mathbf{x}$, the time derivative of the scalar $\mathbf{x}^T\mathbf{M}\mathbf{x}$ can be written in the given form.
4. For a given system of differential equations:
   - Find equilibria points;
   - Given a Lyapunov function, analyze system stability using Lyapunov's direct method;
   - Analyze system stability in a given range.

**Typical questions for seminar classes (labs) within this section**

1. For a given differential equation, find the values of coefficients $k_1$ and $k_2$ that would make it critically damped.

2. Convert a given system of differential equations into state-space form.
3. For an equation given in the state space form, write it as an ordinary differential equation for specified input and output variables.
4. Analyze system's stability given equation of its full energy.
5. Apply Lyapunov's direct method to analyze stability of a physical system.

**Test questions for final assessment in this section**

1. Evaluate stability of a linear system whose dynamics is written in the state-space form.
2. What must be the properties of Lyapunov function candidate and its time derivative to confirm
    - Local stability,
    - Global stability of the system.
3. How to analyze system's stability based on its phase portrait (with examples)?
4. Describe physical motivation behind Lyapunov's direct method and how it used to analyze stability of dynamical systems.

#### N.2.2.3 Section 3

**Section title:** Feedback control systems

**Topics covered in this section:**

- Feedback and building control loops
- Stabilization and trajectory tracking
- Linear regulators (P, PD, PID)

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 0 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Give an example of using feedback in activities of daily life
2. Drive error dynamics equations for a given feedback control law
3. What are the physical analogies for the individual terms of PD-controller?
4. How to implement PD regulator in MATLAB software?

**Typical questions for seminar classes (labs) within this section**

1. Solve numerically in MATLAB a second-order ODE for the following controller types: P, PD, PID.
2. Analyze stability of a given linear control system.
3. How individual gains of PD and PID controllers affect transient and steady-state response?
4. How does underestimation of system parameters affect performance of linear controllers and how to improve it?

**Test questions for final assessment in this section**

1. What is the physical analog of PD-regulator in application to control over second-oder mechanical systems?
2. For a given system described by differential equations, design a linear control system and analyze its stability.
3. Describe pros and cons of linear controllers in application to nonlinear system control.

#### N.2.2.4 Section 4

**Section title:** Feedback linearization

**Topics covered in this section:**

- Joint-space inverse dynamics of serial manipulators
- Stabilization and trajectory tracking problems
- Input-state linearization

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1. For a given differential equation that describes pendulum dynamicsm do the following:
   - Find control law transforming original dynamics into that of a linear mass-spring-damper system;
   - Write position error dynamics for the designed control law (inverse dynamics);
   - Repeat the previous steps if there are uncertainties in some of the system's parameters.
2. For a given general form of dynamics equation, demonstrate that the following control laws guarantee system stability.
3. Simulate dynamics of given nonlinear system in MATLAB for given control law.
4. Perform input-state linearization for a given system of differential equations.

**Typical questions for seminar classes (labs) within this section**

1. Find control law that linearizes a given differential equation.
2. Analyze stability of given nonlinear systems and contribution of individual terms to system stability.
3. Find general feedback linearization law for a given system of differential equations.
4. Analyze stability and limitations of a given feedback linearization control law over a two-link manipulator.

**Test questions for final assessment in this section**

1. What are the pros and cons of feedback linearization approach?

2. Provide examples of systems (differential equations) for which feedback linearization can result in infinite control effort.
3. What are the typical issues when applying feedback linearization approach to control over robotic manipulators?

#### N.2.2.5 Section 5

**Section title:** Robust control

**Topics covered in this section:**

- Sliding modes in dynamic systems
- Robust control in scalar and matrix form
- Stability and tuning of robust controllers
- Control law smoothening.

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 0 |
| Reports | 1 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Name uncertainties typical for mechanical systems
2. Analyze system behavior in sliding mode
3. Simulate system behavior with robust control in MATLAB
4. Numerically implement control law smoothening for a robust controller.

**Typical questions for seminar classes (labs) within this section**

1. How to synthesize robust controller for a given range of system parameters' deviations?
2. Describe the effect of robust gain coefficient $k$ on system stability, control output and resulting system behavior.

3. Design robust controller for a dynamical system described by a given set of differential equations and implement it in MATLAB.
4. Analyze stability of robust control system for a robotic manipulator with given equation of motion.

**Test questions for final assessment in this section**

1. Describe typical sources of uncertainties and parameter deviations in models of physical and mechanical systems, their typical ranges and influence on system behavior.
2. Name pros and cons of robust control systems with and without control law smoothening.
3. How to tune robust controller terms to compensate for system uncertainties?

## N.3 Autonomous Systems

- **Course name:** Autonomous Systems
- **Course number:**
- **Subject area:** Software engineering, Control Theory, Navigation, Motion Planning, Computer Vision

### N.3.1 Course Characteristics

#### N.3.1.1 Key concepts of the class

- Development of algorithms and software for self-driving vehicles
- Interaction with real-scale sensors and computational platforms
- Testing and validation of algorithms and software
- Navigation and planning
- Obstacle avoidance
- Face the self-driving vehicles industry challenges

#### N.3.1.2 What is the purpose of this course?

Students will be introduced to the terminology, concepts, design considerations and safety assessment of self-driving cars. We also will consider different types of sensors such as GNSS, IMU, camera, radar and LIDAR and how they can assist us in state estimation and localization of a self-driving car. Main perception issues and approaches are also considered: static and dynamic object detection, road segmentation and how to localize a self-driving vehicle on the road.

#### N.3.1.3 Course objectives based on Bloom's taxonomy

**- What should a student remember at the end of the course?**

By the end of the course, the students should be able to

- List basic and auxiliary modules of advanced control systems of vehicles,
- Outline advanced approaches to the development of autonomous vehicles,
- Define engineering tasks to solve in order to develop an autonomous vehicle,

- Name sensors commonly adopted for usage in autonomous vehicles,
- Identify the navigation problem for self-driving vehicles,
- Identify relationship between requirements to autonomous vehicles and their hardware and software,
- Name basic sources of information about autonomous vehicles.

**- What should a student be able to understand at the end of the course?**

By the end of the course, the students should be able to

- Choose appropriate technologies to solution of development tasks,
- Understand most important sources of uncertainties related to autonomous navigation and control,
- Understand the issues in proving of self-driving vehicles safety,
- Understand the safety frameworks and current industry practices for vehicle development,
- Understand the key methods for parameter and state estimation used for autonomous driving,
- Understand modern approaches for object detection and road segmentation tasks,
- Describe the most significant algorithms for path planning,
- Understand methods for sensor data analysis and validation,
- Interpret technical requirements as a basis for an autonomous vehicle architecture,
- Describe common plan of actions necessary to achieve desired properties of an autonomous vehicle,
- Name proc and cons and testing and validation of results by simulations and hardware test-benches.

**- What should a student be able to apply at the end of the course?**

By the end of the course, the students should be able to ...

- Develop basic control and planning modules for autonomous vehicles,
- Apply basic algorithms for sensors data analysis and fusion,

- Perform intrinsic and extrinsic camera calibration
- Develop basic navigation modules for autonomous vehicles, implementing different types of Kalman Filter and Particle Filter
- Develop basic perception modules for autonomous vehicles for road segmentation problem,
- Illustrate the main path-planning algorithms for self-driving vehicles navigation problem,
- Establish software stack for development, validation and testing of autonomous vehicle modules.
- Detect, describe and match image features and design his/her own convolutional neural networks.

#### N.3.1.4 Course evaluation

**Table N.7:** Course grade breakdown

| Type | Default points | Proposed points |
|---|---|---|
| Labs/seminar classes | | 10 |
| Homework assignments | | 20 |
| Quizzes | | 10 |
| Interim performance assessment | | 30 |
| Term project | | 15 |
| Exams | | 15 |

If necessary, please indicate freely your course's features in terms of students' performance assessment: n/a.

#### N.3.1.5 Grades range

**Table N.8:** Course grading range

| Grade | Default range | Proposed range |
|---|---|---|
| A. Excellent | 90-100 | |
| B. Good | 75-89 | |
| C. Satisfactory | 60-74 | |
| D. Poor | 0-59 | |

If necessary, please indicate freely your course's grading features.

#### N.3.1.6 Resources and reference material

Main textbook:

- "Mechatronics," Sabri Cetinkunt. John Wiley & Sons., 2007.

Other reference material:

- SAE website `https://www.sae.org/standards/content/j3016_201806/`
- "Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering" (6th Edition), William Bolton. Pearson Education, 2003.

### N.3.2 Course Sections

#### N.3.2.1 Glossary of terms

- **ACC**: Adaptive Cruise Control. A cruise control system for vehicles which controls longitudinal speed. ACC can maintain a desired reference speed or adjust its speed accordingly to maintain safe driving distances to other vehicles.
- **Ego**. A term to express the notion of self, which is used to refer to the vehicle being controlled autonomously, as opposed to other vehicles or objects in the scene. It is most often used in the form ego-vehicle, meaning the self-vehicle.
- **FMEA**: Failure Mode and Effects Analysis. A bottom up approach of failure analysis which examines individual causes and determines their effects on the higher level system.
- **GNSS**: Global Navigation Satellite System. A generic term for all satellite systems which provide position estimation. The Global Positioning System (GPS) made by the United States is a type of GNSS. Another example is the Russian made GLONASS (Globalnaya Navigazionnaya Sputnikovaya Sistema).
- **HAZOP**: Hazard and Operability Study. A variation of FMEA (Failure Mode and Effects Analysis) which uses guide words to brainstorm over sets of possible failures that can arise.
- **IMU**: Inertial Measurement Unit. A sensor device consisting of an accelerometer and a gyroscope. The IMU is used to measure vehicle acceleration and angular velocity, and its data can be fused with other sensors for state estimation.

- **LIDAR**: Light Detection and Ranging. A type of sensor which detects range by transmitting light and measuring return time and shifts of the reflected signal.
- **LTI**: Linear Time Invariant. A linear system whose dynamics do not change with time. For example, a car using the unicycle model is a LTI system. If the model includes the tires degrading over time (and changing the vehicle dynamics), then the system would no longer be LTI.
- **LQR**: Linear Quadratic Regulation. A method of control utilizing full state feedback. The method seeks to optimize a quadratic cost function dependent on the state and control input.
- **MPC**: Model Predictive Control. A method of control whose control input optimizes a user defined cost function over a finite time horizon. A common form of MPC is finite horizon LQR (linear quadratic regulation).
- **NHTSA**: National Highway Traffic Safety Administration. An agency of the Executive Branch of the U.S. government who has developed a 12-part framework to structure safety assessment for autonomous driving. The framework can be found here. $https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf$
- **ODD**: Operational Design Domain. The set of conditions under which a given system is designed to function. For example, a self driving car can have a control system designed for driving in urban environments, and another for driving on the highway.
- **OEDR**: Object and Event Detection and Response. The ability to detect objects and events that immediately affect the driving task, and to react to them appropriately.
- **PID**: Proportional Integral Derivative Control. A common method of control defined by 3 gains.
  1) A proportional gain which scales the control output based on the amount of the error
  2) An integral gain which scales the control output based on the amount of accumulated error
  3) A derivative gain which scales the control output based on the error rate of change
- **RADAR**: Radio Detection And Ranging. A type of sensor which detects range

and movement by transmitting radio waves and measuring return time and shifts of the reflected signal.

- **SONAR**: Sound Navigation And Ranging. A type of sensor which detects range and movement by transmitting sound waves and measuring return time and shifts of the reflected signal.

The main sections of the course and approximate hour distribution between them is as follows:

**Table N.9:** Course Sections

| Section | Section Title | Teaching Hours |
|---|---|---|
| 1 | Introduction to Self-driving cars | 6 |
| 2 | Control systems | 6 |
| 3 | State estimation | 6 |
| 4 | Visual perception | 6 |
| 5 | Motion planning | 6 |

#### N.3.2.2 Section 1

**Section title:** Introduction to Self-driving cars

**Topics covered in this section:**

- Taxonomy of Driving
- Requirements for Perception
- Driving Decisions and Actions
- Sensors and Computing Hardware
- Hardware Configuration Design
- Software Architecture
- Environment Representation
- Safety Assurance for Self-Driving Vehicles
- Industry Methods for Safety Assurance and Testing
- Safety Frameworks for Self-Driving

**Typical questions for ongoing performance evaluation within this section**
**What forms of evaluation were used to test students' performance in this section?**

1.

1. Describe levels of autonomous driving.

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

2. What are the tasks of perception.
3. Name the main sensors necessary for autonomous driving.
4. Discuss main road scenarios.

**Typical questions for seminar classes (labs) within this section**

1. Consider scenario when you are at home and need to drive to work:
   - Name the OEDR tasks.
   - Which of the tasks are associated with perception?
   - Which of the tasks are associated with predictive planning?
   - Determine off-road and on-road objects.
   - Suppose your vehicle has lane keeping assistance, which of the objects are relevant for its performance?
2. Your autonomous vehicle is driving on the German autobahn at 150 km/h and you wish to maintain safe following distances with other vehicles. Assuming a safe following distance of 2s, what is the distance (in m) required between vehicles? Round your answer to 2 decimal places.
3. Using the same speed of 150 km/h, what is the braking distance (in m) required for emergency stops? Assume an aggressive deceleration of $5m/s^2$.
4. According to the report by Rand Corporation, autonomous driving of 8.8 billion miles is required to demonstrate human-level fatality rate of an autonomous vehicle fleet using a 0.95. Confidence Interval. How many years is required to perform this testing with a fleet of 100 vehicles running 24 hours a day, 7 days a week at an average of 25 miles per hour?
5. Given that an autonomous vehicle failure has happened and based on this tree, what is the probability that the failure happened because of Vehicle Control Algorithm Failure OR Inadequate Car Drivers?

6. Given that the autonomous vehicle failure has happened, and based on this tree, what is the probability that the failure happened because of Software Failure AND Extreme Weather Conditions at the same time?
7. There are failures listed below. Which failures should we focus on solving first according to FMEA?

**Test questions for final assessment in this section**

1. Here are some rules for driving at a stop sign. Which of the following is an appropriate priority ranking?
2. Which of the following sensors are used for the lane keeping assistance?
3. You are on the highway and you see a truck in front of you. Assume the car is driving on the right-hand side of the road. There is also a blue car beside the truck in the other lane.
   - Your vehicle follows the truck and maintains a constant distance away. What kind of control is this?
   - You decide to change lanes to pass a truck. What kind of decision is this?
   - Which of the following tasks are rule-based planning?
   - Suppose the blue vehicle suddenly brakes and you decide to abort the lane change. If your vehicle can respond automatically and remain in its own lane, what is the minimum level of autonomy of your vehicle?
   - The blue vehicle returns to normal speed and you can now safely change lanes. Your car is performing the lane change, what kind of control is this?
4. You're finished work and need to drive back home, but it's nighttime. You plan a new path home on your GPS application to avoid the construction site, what type of planning is this?
5. Your new path goes through a school zone and you see the school zone sign. You decide to slow down despite there being no pedestrians or children (it's nighttime). What sort of planning is this?
6. What are the differences between exteroceptive sensors and proprioceptive sensors?
7. Which of the following exteroceptive sensors would you use in harsh sunlight?
8. Why is synchronization and timing accuracy important in the self driving sys-

tem?

9. Suppose your vehicle was using long range cameras for sensing forward distance, but it is now nighttime and the images captured are too dark. Which of the following sensors can be used to compensate?
10. What are the differences between an occupancy grid and a localization map?
11. The vehicle steps through the software architecture and arrives at the controller stage. What information is required for the controller to output its commands to the vehicle?
12. What is (are) the role(s) of the system supervisor?
13. Which of the following tasks should be assigned to the local planner?
14. What common objects in the environment appear in the occupancy grid?
15. Which of the following maps contain roadway speed limits?
16. Which from the below options is the most ACCURATE and COMPLETE definition of risk in terms of self-driving vehicles?
17. Name the major components of an autonomous driving system
18. What are the most common categories of autonomous vehicle hazard sources?
19. Which categories are included in the safety framework to structure safety assessment for autonomous driving defined by NHTSA?
20. Which actions are needed to be performed in the event of an accident by an autonomous vehicle?
21. What are the most common accident scenarios?
22. What kind of safety system is described by the following definition? This system can be analyzed to define quantifiable safety performance based on critical assessment of various scenarios.
23. Which of the following options is the most ACCURATE and COMPLETE definition of functional safety in terms of self-driving vehicles?

#### N.3.2.3 Section 2

**Section title:** Control systems

**Topics covered in this section:**

- Kinematic Modeling in 2D
- The Kinematic Bicycle Model

- Dynamic Modeling in 2D
- Longitudinal Vehicle Modeling
- Lateral Dynamics of Bicycle Model
- Vehicle Actuation
- Tire Slip and Modeling
- Proportional-Integral-Derivative (PID) Control
- Longitudinal Speed Control with PID
- Feedforward Speed Control
- Introduction to Lateral Vehicle Control
- Geometric Lateral Control - Pure Pursuit
- Geometric Lateral Control - Stanley
- Advanced Steering Control - MPC
- Self-Driving Car Simulations

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What are the main approaches for control systems development?
2. Describe the main blocks of longitudinal and lateral control.
3. What are the levels of self-driving car control systems?
4. What are the main forces acting on a car during its movement?
5. What is the most ACCURATE and PRECISE definition of the crosstrack error?
6. What vehicle reference frame is used in a pure pursuit controller?
7. What are major components of the Stanley controller?
8. What is MPC controller?

**Typical questions for seminar classes (labs) within this section**

1. Implement the kinematic bicycle model. The model accepts velocity and steering rate inputs and steps through the bicycle kinematic equations. Once the model is implemented, students will provide a set of inputs to drive the bicycle in a figure 8 trajectory.
2. Implement the forward longitudinal vehicle model. The model accepts throttle inputs and steps through the longitudinal dynamic equations. Once implemented, students will be given a set of inputs that drives over a small road slope to test your model.
3. A vehicle is being operated on a highway with the reference velocity of 126 km/h (35 m/s) in gear 4 and it overcomes the total load torque of 300 ft-lb. This vehicle specification includes effective wheel radius of 0.35 m and $4^{th}$ gear ratio of 2. What throttle angle is required for maintaining the the current speed of the vehicle?
4. Compute the radius from the instantaneous center of rotation to the center of the vehicle rear axle (in m) required for an autonomous vehicle to follow the desired path. The lookahead distance is 10 m. The car length is 4 m. The angle between the vehicle's body heading and the lookahead line is $30^0$.
5. Compute the steering angle (in degrees) required for an autonomous vehicle with pure pursuit lateral control for following the desired path. The lookahead distance is 15 m. The car length is 5 m. The angle between the vehicle's body heading and the lookahead line is 60°.
6. Consider a situation in which a vehicle traveling speed has decreased from 100 km/h to 50 km/h. This vehicle lateral control is implemented with a pure pursuit controller where $l_d$ is assigned as a function of vehicle speed. How should $l_d$ change in this situation?
7. What is the correct figure of the crosstrack error dynamics for a small error value (where $e(t) = -k * e(t)))$?

**Test questions for final assessment in this section**

1. Determine the order of the provided transfer function.
2. Determine the poles and zeros of the provided transfer function

3. What might be your action as a system control engineer if you need to increase the overshoot of a control loop system?
4. As a system control engineer, you constructed the provided closed loop transfer function to represent the Mass-Spring-Damper System. What is the correct transfer function for this closed loop?
5. Students are given the step response of a few different PID controllers using the same gains for the same first order transfer function. Determine a possible set of controllers that generated these step responses.
6. What is the output of a typical output of a Longitudinal control module?
7. Based on the engine map in the provided figure, determine the throttle angle needed to produce 250 ft-lb of torque given that the current engine speed is 3500 RPM.
8. The results of a simulation of the control response to a step change in desired speed of a dynamic vehicle model with a PID controller are shown in the provided figures . There are two spikes on these figures: one spike is between 2 and 3 seconds, another spike is between 3 and 4 seconds. What is the reason of these spikes?
9. What type of control system is shown in the provided figure?
10. What types of inaccuracies are corrected by a feedback controller?
11. What assumptions are essential for creation of a longitudinal feedforward input?
12. What are the sources of the load torque considered for a longitudinal feedforward look-up table computation?
13. Which reference path is the most compact and easy to construct?
14. What is the value of the crosstrack error, governed by the ODE $e'(t) = -ke(t)$, at $t=2$, given that $e(0)=4$ and $k=1$?
15. What is the typical way of finding the solution for a nonlinear vehicle dynamics model given an input function?
16. What is the output of the Model Predictive Controller described in this course?

#### N.3.2.4 Section 3

**Section title:** State Estimation

**Topics covered in this section:**

- Squared Error Criterion and the Method of Least Squares
- Recursive Least Squares
- Least Squares and the Method of Maximum Likelihood
- The (Linear) Kalman Filter
- Kalman Filter and The Bias BLUEs
- Going Nonlinear - The Extended Kalman Filter
- An Improved EKF - The Error State Extended Kalman Filter
- Limitations of the EKF
- An Alternative to the EKF - The Unscented Kalman Filter
- 3D Geometry and Reference Frames
- The Inertial Measurement Unit (IMU)
- The Global Navigation Satellite Systems (GNSS)
- Light Detection and Ranging Sensors
- LIDAR Sensor Models and Point Clouds
- Pose Estimation from LIDAR Data
- State Estimation in Practice
- Multisensor Fusion for State Estimation
- Sensor Calibration - A Necessary Evil
- Loss of One or More Sensors

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 0 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. In which of the cases would the method of weighted least squares produce valid solutions
2. Describe the method of maximum likelihood
3. Implement your own recursive estimator. Implement an Extended Kalman filter, which will estimate the trajectory of a vehicle using odometry, range and bearing

measurements.
4. Describe rotation matrix
5. Describe GPS satellite signals

**Typical questions for seminar classes (labs) within this section**

1. Implement the batch least squares method.
2. Convert your batch least squares solution to a recursive solution.
3. You are measuring the voltage drop $V$ across an electrical component using two different multimeters; one of the meters is known to be more reliable than the other. Which method would you use to estimate the best voltage value from noisy measurements?
4. Measurements are drawn from a Gaussian distribution with variance $\sigma^2$. Which of the estimators will provide the 'best' estimate of the true value of a parameter?
5. A single 3D LIDAR reading consists of elevation, azimuth and range measurements $(\epsilon, \alpha, r)$=$(5^0, 10^0, 4m)$. Assuming that the measurements are noiseless, calculate the position of this point in the Cartesian sensor frame. Note that the elevation and azimuth angles are given in degrees (for convenience) - you will need to convert these values to radians for use with most trigonometric functions.
6. A 3D LIDAR unit is scanning a surface that is approximately planar, returning range, elevation and azimuth measurements. In order to estimate the equation of the surface in parametric form (as a plane), we need to find a set of parameters that best fit the measurements. Implement functions, which transform LIDAR measurements into a Cartesian coordinate frame and estimate the plane parameters, respectively. You may assume that measurement noise is negligible. The code comments provide more information on the format of the arguments to each function.
7. You are testing an algorithm for LIDAR-based localization on a vehicle in a controlled environment. What are some of the things you need to take into account?

**Test questions for final assessment in this section**

1. Given the above histogram of noisy measurements, it is appropriate to use a LS estimator?
2. Looking at the histogram in the previous question, what could be the reason for such a distribution of measurements?
3. Which of provided rotation matrices are valid?
4. Localization can be performed on board a vehicle by integrating the rotational velocities and linear accelerations measured by an IMU. Assuming that the IMU measurement noise is drawn from a normal distribution, what will the pose estimation error look like?
5. What is the minimum number of GPS satellites required to estimate the 3D position of a vehicle through trilateration
6. To estimate the motion of a self-driving car, it is necessary to transform LIDAR scan points from the sensor frame to the vehicle frame. The rotation of the vehicle frame with respect to the LIDAR frame is represented by the rotation matrix $C_{vl}$. Given any point $p_l^i$ in the LIDAR frame, and considering rotation only, which of these expressions correctly transforms the point into the vehicle frame?
7. Implement the Error-State Extended Kalman Filter (ES-EKF) to localize a vehicle using data from the simulator

#### N.3.2.5 Section 4

**Section title:** Visual perception

**Topics covered in this section:**

- The Camera Sensor
- Camera Projective Geometry
- Camera Calibration
- Visual Depth Perception - Stereopsis
- Visual Depth Perception - Computing the Disparity
- Image Filtering
- Introduction to Image features and Feature Detectors
- Feature Descriptors
- Feature Matching

- Feature Matching: Handling Ambiguity in Matching
- Outlier Rejection
- Visual Odometry
- Feed Forward Neural Networks
- Output Layers and Loss Functions
- Neural Network Training with Gradient Descent
- Data Splits and Neural Network Performance Evaluation
- Neural Network Regularization
- Convolutional Neural Networks
- The Object Detection Problem
- 2D Object detection with Convolutional Neural Networks
- Training vs. Inference
- Using 2D Object Detectors for Self-Driving Cars
- The Semantic Segmentation Problem
- ConvNets for Semantic Segmentation
- Semantic Segmentation for Road Scene Understanding

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. What is the definition of the camera obscura?
2. Why is camera calibration important in the self-driving car domain?
3. What methods from linear algebra can we use for solving camera calibration problem?
4. What are some DISADVANTAGES of this linear calibration model?
5. Name the main building blocks of a machine learning system
6. What are the strategies are used for regularization in neural networks?
7. What the reasons for the difficulty in performing object detection?

**Typical questions for seminar classes (labs) within this section**

1. Apply Stereo Depth to a Driving Scenario
2. Imagine a situation in which a camera mounted on a car sees a point O on a sign-post. The location of this point in world coordinate system is $O = [-0.5, 1.5, 9]^T$ meters. The location of the center of the world coordinate system relatively to the camera optical center in camera frame coordinates is defined by the translation vector $M = \left[1, 2, 10\right]^T$ meters and 180° rotation around the $X_C$ axis. The camera intrinsic parameter matrix is $K = \begin{bmatrix} 640 & 0 & 640 \\ 0 & 480 & 480 \\ 0 & 0 & 1 \end{bmatrix}$ and the image resolution is 1280×960 pixels. What is the position of this point in the camera coordinate system?
3. Based on the problem presented in the above question, what is the pixel location of the 2D projection of the point O on the image plane?
4. Implement visual odometry for localization in autonomous driving
5. A feedforward neural network has an input layer, 5 hidden layers and an output layer. What is the depth of this neural network?
6. Implement the ReLU activation function using numpy
7. Which output unit/loss function pair is usually used for regression tasks that use neural networks?
8. Which of the providing conditions might be used as a stopping condition for gradient descent
9. The input to a pooling layer has a width, height and depth of 224x224x3 respectively. The pooling layer has the following properties:
   Kernel shape: 2x2 Stride: 2 What is the width of the output of this pooling layer?
10. A semantic segmentation model uses the provided decoder architecture. The convolutions are all 3x3, have a padding size of 1, and have a number of filters shown in the figure. The up-sampling multiplier S is 2 for all upsampling layers.

**Test questions for final assessment in this section**

1. If the baseline between camera centers is known for a stereo rig, what limitation of monocular vision can be avoided?

2. Consider a stereo camera setup in the provided figures. Which of the statement about this configuration are correct?
3. What parameters and computations are needed to perform depth calculations from disparity measurements?
4. A naive solution for the stereo correspondence problem is an exhaustive search, where we search the whole right image for a match to every pixel in the left image. Why is this a bad approach?
5. What is the definition of an epipolar line for stereo cameras?
6. Which of provided 3X3 image filters is a Gaussian filter?
7. How are neural network bias parameters usually initialized at the beginning of training?
8. You are working on a self-driving car project and want to train a neural network to perform traffic sign classification. You collect images with corresponding traffic sign labels, and want to determine the number of frames you will use for training. Given that you have around one million images with labels, what training/validation/testing data split would you use?
9. You finish training your traffic sign classifier, and want to evaluate its performance. You compute the classification accuracy on the training, validation, and testing data splits and get the provided results. You know that a human has an accuracy of around 0.98 on the traffic sign classification task. What are things you might try to achieve human level performance?
10. You are a self-driving car perception engineer developing an object detector for your self-driving car. You know that for your object detector to be reliable enough to deploy on a self-driving car, it should have a minimum precision of 0.99 and a minimum recall of 0.9. The precision and recall are to be computed at a score threshold of 0.9 and at an IOU threshold of 0.7. You compute the IOU of your detector on a frame with ground truth with the provided results. Assuming that the single frame is sufficient to characterize the performance of the object detector, is your system reliable to be used on a self-driving car?
11. The input to a convolutional layer has a width, height and depth of 224x224x3 respectively. The convolutional layer has the following properties:
    Kernel shape: 3x3x256 Stride: 2 Padding: 3 What is the depth of the output of this convolutional layer?

12. What type of output layer is most commonly used in the regression head of a convolutional object detector?
13. Achieving smooth category boundaries is a major difficulty to take into account while designing semantic segmentation models. Which of the provided statements describe the origins of this problem?
14. When comparing the results of a semantic segmentation model to the ground truth, you found out that for the car category, its class IOU is 0.75. Knowing that the number of false positives (FP) is 17, and the number of false negatives (FN) is 3, what is the number of true positives achieved by this model?
15. Which of the provided statements do you typically see in a Semantic Segmentation Model?
16. Which of loss functions is usually used to train semantic segmentation models?
17. Which of the provided categories in a semantic segmentation output map would be useful to determine lane boundaries?
18. How many points are needed to estimate a plane model?
19. What techniques can be used to estimate lines that could belong to lanes in a post-processed output image from semantic segmentation?

#### N.3.2.6 Section 4

**Section title:** Motion planning

**Topics covered in this section:**

- Driving Missions, Scenarios, and Behaviour
- Motion Planning Constraints
- Objective Functions for Autonomous Driving
- Hierarchical Motion Planning
- Occupancy Grids
- Populating Occupancy Grids from LIDAR Scan Data
- Occupancy Grid Updates for Self-Driving Cars
- High Definition Road Maps
- Creating a Road Network Graph
- Dijkstra's Shortest Path Search
- A* Shortest Path Search

- Motion Prediction
- Map-Aware Motion Prediction
- Time to Collision
- Behaviour Planning
- Handling an Intersection Scenario Without Dynamic Objects
- Handling an Intersection Scenario with Dynamic Objects
- Handling Multiple Scenarios
- Advanced Methods for Behaviour Planning
- Trajectory Propagation
- Collision Checking
- Trajectory Rollout Algorithm
- Dynamic Windowing
- Parametric Curves
- Path Planning Optimization
- Optimization in Python
- Conformal Lattice Planning
- Velocity Profile Generation

**What forms of evaluation were used to test students' performance in this section?**

| Form | Yes/No |
|---|---|
| Development of individual parts of software product code | 0 |
| Homework and group projects | 1 |
| Midterm evaluation | 0 |
| Testing (written or computer based) | 1 |
| Reports | 0 |
| Essays | 0 |
| Oral polls | 1 |
| Discussions | 1 |

**Typical questions for ongoing performance evaluation within this section**

1. Which are examples of common scenarios in the autonomous driving motion planning problem?
2. What are some examples of dynamic obstacles?
3. What is a drawback of using a sampling-based method for path planning?
4. Describe the mission underlying an autonomous vehicle's mission planner
5. What is a graph in the mission planning context?

**Typical questions for seminar classes (labs) within this section**

1. True or false, the autonomous driving mission takes pedestrian behaviour into consideration.
2. True or false, "Staying Stopped" is a maneuver that is useful for handling traffic light controlled intersections.
3. What are the reasons for decomposing motion planning into a hierarchy of optimization problems?
4. True or false, instantaneous curvature is the inverse of the instantaneous turning radius at a point on a curve.
5. True or false, the time gap is the amount of time that it would take for the ego vehicle to reach the current position of its leading vehicle.
6. True or false, the friction ellipse is always a tighter constraint than the comfort rectangle.
7. True or false, jerk is the derivative of acceleration with respect to time.
8. True or false, mission planning focuses on map-level navigation from the ego vehicle's current position to a final destination.
9. True or false, a conformal lattice planner selects goal points ahead of the car that are laterally offset from the centerline of the road, plans paths to each goal point, then selects the best collision-free path according to some objective function.
10. Generate an occupancy grid using lidar scanner measurements from a moving vehicle in an unknown environment
11. Implement Dijkstra's search algorithm on a road network in Berkeley, California. You will then modify that algorithm using a distance heuristic to perform A* search.
12. True or false, Breadth-First Search (BFS) will explore the graph using a "last-in-first-out" data structure known as a stack.
13. True or false, Breadth-First Search (BFS) will always find the optimal (shortest) path in an unweighted graph.
14. True or false, Breadth-First Search (BFS) will always find the optimal (shortest) path in a weighted graph.
15. True or false, in a min heap, the root of the heap (the first element) contains the node with the smallest value.

**Test questions for final assessment in this section**

1. To generate the shortest path to a point, we need to minimize...
2. The integral of difference (IOD) term in a planning objective function can be used to...
3. What are some examples of the inputs a finite state machine might take in the context of behaviour planning for autonomous driving?
4. True or false, reinforcement learning relies on interacting with an environment during the learning
5. In these graph search algorithms, what is the main purpose of keeping track of a "closed" set of graph vertices?
6. What is a min heap data structure?
7. In Dijkstra's algorithm, suppose during the process of adding vertices to the open set, we come across a vertex that has already been added to the open set. However, this time we have found a lower cost to reach this vertex than is presently stored in the open set's min heap. What should be done?
8. What is a search heuristic in the context of mission planning?
9. Suppose I have a vertex at location (2.0, 3.0) and another at location (4.0, 5.0). What is the Euclidean distance between these two points (to three decimal places)?
10. True or false, an admissable heuristic to the A* search algorithm will never underestimate the cost to reach the goal vertex.

# Appendix O

# Catalogue of the core courses