–

Innopolis University

University Academic Council

OVERALL TEACHING REGULATIONS AY 2019/2020

February 8, 2021

Curriculum Committee

1st February 2020

Contents

Part I

Overall Regulations

1 About this document

This document is the overarching document summarizing the educational process in place at Innopolis University.

1.1 How to update this document

This document is permanently updated by a permanent committee working under the direction of the head of the Department of Education and of the Associate Dean for Teaching of the Faculty of Computer Science and Engineering.

1.2 Final version of the document for the academic year

One week before the beginning of the academic year the head of the department of education "freezes" the current version of this document, signs it, stores it in the official document repository system. This version of the document is considered the current version for such academic year.

2 Innopolis University Manifesto on Medium of Instruction

Innopolis University is committed to the highest international academic standards and offers the opportunity to study Computer Science and Engineering in English developing fluency and command of technical and non-technical vocabulary. This is made possible by the University's international community of staff and students, speaking several foreign languages, and in particular English, on a daily basis: students, professors, researchers and administrative staff.

This commitment is reflected on using English as the language of instruction of Innopolis University. All classes, teaching, labs, assignments, exercises and practical material are provided in English. English is expected to be used in the following situations:

- Frontal Classes, including questions
- Frontal Labs, including questions
- Class and labs material
- Technical and non-technical assignments
- Theses and projects
- Written and Oral Exams
- Group Office Hours
- Public Chats related to University life
- Emails to students or colleagues
- Student's Groups
- Any other document related to education

Being English the official Medium of Instruction of Innopolis University does not prevent the use of other languages, and, in particular, Russian, in other situations where two or more people share a common language, thus English is not necessary used in situations like informal conversation (questions after classes...), individual office hours, private emails, private chats, ...

2.1 Rationale

Fluency in multiple languages reflects on one hand a versatile mind, on the other hand the significant effort and dedication necessary to acquire such ability. Being multilingual is a notable skill, no matter what set of languages is in a student's portfolio, however English is an important addition for every graduate. The importance of English language can be summarized in three major points:

- English as a communication means
- English as language of employment
- English as language of science

2.1.1 English as a communication means

English is the language of international communication; speaking English increases the chances to communicate virtually anywhere in the world in every field of knowledge. Fluency in this language allows graduate to travel, encounter new cultures, ideas and to grow as individual and professionals. It is the medium for exchanging technical knowledge as well feelings and emotions.

2.1.2 English as language of employment

The opportunities available on the job market today requires graduates to manage a complex mix of soft and hard skills. Among these language skills have a primarily importance, not only for international job market, but also for seeking high-profile employment in the Russian Federation. All the major companies keep indeed continuous relations with abroad institution and engineers, manager, salesmen and every other highly-qualified specialists needs the command of at least one foreign language, typically English. English skills are just as desirable to employers in any country as they are to employers in English-speaking countries. Fluency in English allows employees to attend and organize international business meetings. Where several languages are represented, the chances are that the meeting will be conducted in English. Employees with language skills will find themselves in a position of advantage and will be able attending relevant gathering, therefore having the opportunity to advance faster in the career ladder.

2.1.3 English as language of science

English is also the language of science and being fluent allows graduates to continuously educate themselves accessing scientific literature and attending international conferences. English is essential to pursue a scientific career and to develop the necessary network able to develop your scientific research.

2.2 Commitment to improve language skills

In order to gain admission to Innopolis University, it is necessary to provide evidence of specific language levels based on the specific program of enrollment. Over the period study the English level is expected to grow. The University implements several ways to support students in their path, some of them being:

- English Courses
- Communication Courses
- Technical Essays in English
- Lab internships
- International projects

2.3 Exceptions

Specific exceptions can be made by the Department of Education for courses that need to comply with laws and regulations of the Russian Federation (for example “Safety”). The University commits to provide the opportunity to learn the material of these courses to all the international students.

In some cases, instructors who are fluent in Russian and possess the appropriate established terminology used in Russian industry and research can familiarize students with bilingual terminology that could give students a competitive advantage in the knowledge of modern terms used at the national and international levels.

2.4 Summary

Innopolis University is committed to provide education of the highest quality using English as a Medium of Instruction to better prepare future graduate to a dynamic job environment.

Staff, teacher and students commit to use English for classes and other major educational situations. Other shared languages can be effectively used in point-to-point communication whenever there is certainty that no involved participants could be excluded by the conversation for lack of fluency in the used language. This chapter of the teaching regulations collects the suggested practices for staff, teacher and student to keep alive the international community and be competitive at the national and international level.

3 ABOUT LECTURING A COURSE

This section presents several aspects involved in the conduction a course, students' assessment, and other administrative tasks.

3.1 Course Structure and Syllabus

Faculty members can use any instructional mode; however, they should always make sure to develop and communicate the objectives of the course to the students. In regards to this, the instructors of the courses should prepare the course syllabus for all of your courses and share them with the Department of Education (DoE) before a semester begins.

The template that IU is currently using for course syllabus is in Appendix P. Such template is then reflected by the DoE into the online teaching infrastructure (currently Moodle) and is used as the core reference for supplying course material, schedules, and calendars to students.

The deadlines to prepare and submit the syllabus are June 30 (for Fall semester courses), October 31 (for Spring semester courses), May 01 (for Summer semester courses).

The instructor of the course is expected to teach with physical presence at least 70% of the lectures.

3.2 Evaluation of students performance

The following is stipulated by the University Regulation "Ongoing and interim assessment of students' knowledge in autonomous noncommercial organization of higher education «Innopolis university»."

3.2.1 Ongoing academic progress assessment (OPA)

1. Ongoing academic progress assessment (OPA) is carried out by all types of self-study and contact work of students prescribed by the disciplines, practices and programs of research work. The OPA is carried out by **professor** giving courses, by the **head of practice** or by the **tutor** of the research work.

2. The forms of the OPA include:
   - individual realization of software code,
   - individual assignments,
   - practical and laboratory work,
   - projects,
   - examinations,
   - colloquiums,
   - tests (hand-written or computer-based),
   - reports, essays, surveys,
   - oral polls,
   - discussions, training, workshops.
3. **Other forms of OPA might be established** in accordance with the curriculum and schedule.
4. The Instructor determines the frequency, the procedure and the forms of OPA, as well as the evaluation system for it. The Instructor has a right to introduce corrections in the quantity and procedure of OPA.
5. The Instructor must **enter OPA results into the LMS Moodle**.
6. The OPA results must be taken into account when the final grade is given during course evaluation. Grades received by students during the OPA are **not subject to retake**.

3.2.2 Students mid-semester evaluation

Within the framework of the quality assessment of the educational process, the IU Department of Education in the middle of each semester analyzes students' progress based on the OPA results. Based on this analysis, employees of the DoE hold special meetings with the under-performing students to assess the reasons behind them struggling. To help this cause,

1. The instructor of each course, at the end of the 8th week of the semester, should estimate of the performances of the students to date, that is, the expected final grade that the student should get if s/he continues with such performances
2. Such estimate should be defined with the usual letter grades (A, B, C, D - where D means "fail")
3. Such estimate should be placed in the Moodle section of the course using an assessment named "Mid Semester Evaluation"

Such mid-of-semester evaluation is **not the midterm evaluation**.

3.2.3 Final evaluation

The official term for the **final course evaluation** is the "Interim assessment" (IA), and it will be used below. The following is stipulated by the University Regulation "Ongoing and interim assessment of students' knowledge in autonomous noncommercial organization of higher education Innopolis university":

1. The **schedule and the place of IA is drafted by the DoE one week prior** to the beginning of an IA, which must be approved by the Director of the University. Academic **faculty and students must be informed of this via their corporate e-mails**.
2. The IA is conducted after completing the course or practice in full or in part. This assessment will take place in a **form that depends on the educational program, the work program of the discipline** (course modules) and following the procedures, prescribed by relevant regulations.
3. The forms of interim assessment are:
   - a test (including a pass/fail test),
   - an examination.

   Both of these can be carried out in the **following forms:**
   - verbally and in a written form,
   - in the form of testing, including the use of computers, software and hardware, as well as
   - in the form of an essay or
   - project defense.

   The **form of the interim assessment is determined by the Instructor**. Students **must be informed about the IA form**.
4. The final grade for the course (module) and practice (research work) can include the results of the OPA.
5. Recording the IA results can be conducted in the following 2 ways:
   - The Instructor enters IA results into the LMS Moodle after which the results are transferred into the intermediate attestation report.
   - The Instructor fills out the intermediate attestation report (Appendix 1 of the cited document) and submits it to the DoE. The DoE employees then transfer the grades into LMS Moodle.
6. The results **cannot be reviewed after being entered into LMS Moodle** and interim assessment report.

7. Final results of the examination (tests) from interim assessment report shall be entered into the student's electronic record books (LMS Moodle, system of education process control *1C: University*, student's portfolio, and student's personal account (`my.university.innopolis.ru`)).

3.2.4 Rules on final examinations

1. If a student has **no valid reason** for being absent from the exam, he/she is considered to not have passed the exam and not fulfilled academic requirements.
2. The Instructor **can provide the list of supportive materials** that students are allowed to use during the exam.
3. The **use of non-allowed supportive materials** during the exams, breach of study discipline (cheating, use of mobile phones or PC, attempts to talk to others, etc.) as well as infringement of the regulations are prohibited. These incidents can serve as a ground for **removal of the student** from the room and **awarding them a *D*** (unsatisfactory) or *F* (Fail) marks.
4. If any of incidents described above occur, the **Instructor should inform the Vice-Rector for Education** of this in written form or via the corporate mail. The Vice-Rector in their turn sends an official note on Regulations infringement to the Director of the University for further consideration and decision.
5. **Presence of the third parties** on exams and tests without the written permission of the Vice-Rector for Education of the University is not allowed.
6. Students with an **individual education plan** may pass tests and exams only during the period determined by their individual education plan.
7. Students with a **valid reason of absence** (illness, family circumstances, etc.) who were not able to pass tests and exams within the established timeframe are permitted to retake an exam on individual terms. The terms are approved by the direct order of the University Director.
8. For students who provide **justification document** for their absence in exam (sick leave sheet, medical certificate, etc.), the examination period may be extended by the number of calendar days indicated in the presented document. This **document should be provided no later than three working days** after being issued.
9. The following procedure is applied for the **extension of the examination session**:
   - A student provides the DoE with: 1) A statement in the name of Vice-Rector for Education to extend the examination period; 2) Documents confirming the

reason for the extension of the examination period (a medical certificate or other documents).
- The Vice-Rector for Education examines the submitted documents and visas student's statement.
- Based on this, the Department of Education of the University drafts submission for the Director on the extension of examination period, indicating a concrete number of extension days.

10. A student is not allowed to have extension of exam period if he/she does not submit the justification document in time.

3.2.5 Marks and grading policy

Table 3.1: Grading rules

Five-point grading system	IU system	Criteria of student's grading
“Excellent”	A	A student gets A (“Excellent”) mark when he/she shows his/her full knowledge of the subject of a study program.
“Good"	B	The learner has the knowledge of the subject in almost full volume of the curriculum; independently, in a logical sequence and in a comprehensive manner, answers all the questions, emphasizing the most essential, is able to analyze, compare, classify, generalize, and systematize the studied material, highlight the main concepts, establish causal relations; clearly formulates answers, freely interprets analytical results and solves situational problems of increased complexity; is well-acquainted with basic literature and research methods; can correlate the theoretical aspects of the activities of a subject with practical problems.
“Satisfactory"	C	The learner has only basic knowledge of the discipline; has difficulties when answering questions, operates with inaccurate wording, misses essential parts of questions. The student is able to solve only the easiest tasks.
“Unsatisfactory"	D	The student did not master the required minimum of subject knowledge, is unable to answer questions even with additional probing questions from the instructor.
“Passed"	P	
“Fail"	F	The student did not master the required minimum of subject knowledge, is unable to answer questions even with additional probing questions from the instructor.

1. A brief summary on grading rules is presented in Table 3.1.
2. The grades A, B, C and P are considered positive and are **not subject for retake in order to improve the grade**. The grades D and F are considered negative (unsatisfactory).
3. A student who have completed in full the requirements of current year's curriculum and have successfully passed all examinations and internships is transferred to the next year in accordance with the Director's order based on the proposal of the Vice-Rector for Education of the University.

3.3 Retakes and missed assignments

The following is stipulated by the University Regulation “Ongoing and interim assessment of students’ knowledge in autonomous noncommercial organization of higher education Innopolis university":

1. The negative results of interim assessment in one or several academic subjects, courses, disciplines (modules), practice (educational, industrial, pre-diploma) of the education program as well as absence on the exam without valid reason are considered as **unfulfilled academic requirements**. Students are required to fulfill these requirements within the timeframe established by the University.
2. The **University defines the schedule** of retakes for each course and practice and designates for these purpose an additional exam (retake) session. This schedule is approved by the director of the university and is **communicated to students through corporate e-mail**.
3. The **structure of the retake** (including labs, oral, written, practical) is defined by the instructor. The grading policy is up to the instructor and needs to be clearly stated in the syllabus.
4. If the **student** has failed to fulfil his/her academic requirement during the retake for the first time (hereinafter - first retake) they **have a right for a second retake with the committee** assigned by the university for this retake.
5. The **first retake should be held by the same instructor** to whom the student failed the exam (test) for the first time. If the instructor does not have the opportunity to conduct the exam at the fixed time, the Vice-Rector for Education appoints another instructor.
6. The **Committee for the second retake is approved by the director’s order** on the basis of the submission of the Vice-Rector for Education of the University. The opportunity to fulfill academic requirements is granted to each student no more than twice.
7. **Student must pass the retake no later than one year** since the date the unfulfilled requirement originated. This period does not include the following: sick leave, academic leave, maternal leave.
8. University has the right to **conduct the first and the second retake during vacation**. In this case, the University must set several retake periods, both during the holidays and during the semester.
9. The **second retake could not be scheduled** to take place **during the internship (practice) period or interim assessment period** (with the exception of the IA in remote learning format).

10. Graduating students are allowed to complete their unfulfilled academic requirements before the order of their admission to the state final certification is issued.
11. The **mark for the second retake is awarded based on agreement of all committee members**. The decision of the committee is adopted by a simple majority of votes, is final and is **not subject to revision**.
12. The results of the retake (corresponding scores and marks) are recorded in accordance with the general rules for entering results of interim assessment. This information is recorded by the instructor in the interim reassessment paper (see Appendix 2, Appendix 3).
13. The students who did not liquidate the academic debt in due time **are expelled** from the University as having failed to fulfill their responsibilities for the conscientious mastering of the educational program and the implementation of the curriculum.
14. The retakes take place during the first week of the following semesters.

3.4 Ongoing and interim assessment for students with disabilities

The following is stipulated by the University Regulation “Ongoing and interim assessment of students’ knowledge in autonomous noncommercial organization of higher education Innopolis university":

1. In order to conduct ongoing and interim assessment for people with disabilities the materials must be drafted. These materials must be adapted to estimate the achievements fixed in the professional study program and the level of achieved competencies in accordance with this study program.
2. For students with disabilities the forms of ongoing and interim assessment must be adapted taking into consideration the peculiarities of their mental and physical development, individual capabilities and health (oral form, written form on paper and computer, in a form of test etc.).
3. In order to make education available for people with disabilities the University must meet the following requirements: the presence of the assistant(s) in the room to provide the disabled people with the necessary technical assistance; written exercises are dictated to the student with disabilities. If necessary, students with disabilities must be allowed to use any technical assistive devices.
4. For those students who have serious health problems such as speech disorder, difficul-

ties with hearing , oral assessments must be converted into a written form.

5. If needed, the disabled student is entitled to request in written form additional time to prepare his/her answers during the test or exam. The duration of the assessment session cannot be extended by more than 1.5 hours.
6. During the assessment sessions, the disabled students must be granted the chance to use assistive devices by taking into consideration their special needs.

3.5 Storage of documents

In order to fulfill the requirements of Russian federal regulations, Department of Education suggests the following policy. Written paper works,

1. created by students at midterm exams, final exams and retakes should be stored by a professor for 1 year from the moment of exam. This category also includes other works that students and/or professors decided to include in student's professional portfolio. Keep these works at professor's office.
2. thesis works, course works and other works with your and student signatures should be stored by DoE. Regardless of the age – take them to Department of Education.
3. created during other activities (even if one year did not pass) can be recycled immediately. You are free to recycle these works with no preliminary actions (burning, shredding, ...).

Comment: the form of exam is defined in syllabus. If an exam is conducted in practical or oral form and produces no paper artefacts, this should be explicitly specified at the course syllabus. Otherwise ministry assumes exam is conducted in written form.

3.6 Transfer Students for BS

These are stipulated by the "Regulations on procedure of transfer, expulsion, and reinstatement of students and grating of academic leaves" from June 1, 2017.

1. Students are allowed to transfer their education program within the first 4 weeks of a semester. The request is extended by the student.
2. The request to change a program is submitted by a student to DoE within the first two weeks of an academic calendar.

3. Within one working week from the day of submission of the request, DoE assesses the eligibility of the student for the requested transfer. In case of a positive evaluation, an interview is scheduled with the Program Coordinator.
4. On the day of the interview, the Program Coordinator holds a 1-3 hour interview and/or an exam with the student to evaluate her/his academic background and motivation.
5. Within the third week of the semester, a meeting is held in which the Program Coordinator, Dean, Vice-Rector and the Head of DoE discuss the group of applicants and make a final decision on their applications.
6. Within one working day from the day of the final decision, DoE provides feedback to students regarding the outcome of their applications.

3.7 Transfer Students for MS

1. Students are allowed to transfer their education program within the first 4 weeks of a semester. The request is extended by the student.
2. The request to change a program is submitted by a student to DoE within the first two weeks of an academic calendar.
3. Within one working week from the day of submission of the request, DoE assesses the eligibility of the student for the requested transfer. In case of a positive evaluation, an interview is scheduled with the Program Coordinator.
4. On the day of the interview, the Program Coordinator holds a 1-3 hour interview and/or an exam with the student to evaluate her/his academic background and motivation.
5. Within the third week of the semester, a meeting is held in which the Program Coordinator, Dean, Vice-Rector and the Head of DoE discuss the group of applicants and make a final decision on their applications.
6. Within one working day from the day of the final decision, DoE provides feedback to students regarding the outcome of their applications.

3.8 Meeting with the Student-Representatives

During the 7th or the 8th week of the semester the instructor must perform a meeting with the student representatives where s/he discusses the evolution of the course, gather feedback from the students, identifies corrective actions, and send a report of such meeting to the education department.

3.9 Management of the Thesis

Thesis are carried out according to the requirements of the Federal law N^{o} 273-FZ of 29.12.2012 and the order N^{o} 636 of the Ministry of education of the Russian Federation of 29.06.2015. Specifically, the following provisions apply to Innopolis University:

1. For the Fall semester,
   (a) all the policies applied to courses are applied to the thesis;
   (b) in the final evaluation in the Fall semester an assessment at the C level should be assimilated to a Fail, since there is an expectation that theses should not be mediocre.
2. For the final (Spring) evaluation of the thesis, the final grade is defined by the state thesis commission, on the basis of a proposal made by the supervisor in Moodle.
3. Details on the implementation of the thesis are in Appendix **??** on page **??** for the BS theses and in Appendix **??** on page **??** for the MS theses.
4. the amount of plagiarism in texts for the thesis of the students enrolled in the BS programs should not exceed 30%;
5. the amount of plagiarism in texts for the thesis of students enrolled in the MS programs should not exceed 25%.
6. a summary report of the inspection of the texts of the thesis of University students in Plagiarism Checking System has to be submitted to the State Examination Committee by the Department of Education no later than 3 (three) calendar days before the day of the thesis defence.
7. theses made by the students of the University have to be published in the electronic library system of the University within 5 (five) days after the defence, except for the works:
   – containing information being a state secret;
   – containing information that has current or potential commercial value (production, technical, economic, organizational) for the rightholder due to it being unknown to third parties, and other data (including the results of intellectual activity in the scientific and technical sphere) that concerns methods of professional activity.

3.10 Participation in lab activities

1. The following provisions do not apply to student developing a thesis or a curricular project course, for which specific regulations are set.
2. During a semester of study, students can participate at activities of a lab of Innopolis University.
3. Such participation is completely voluntary on the side of the student and their hosting is completely voluntary on the side of the accepting professor and it is understood that it is undertaken because both the student and the professor consider it useful for their studies and researchers.
4. Accepting professors are encouraged to set up written agreements with students to clarify the mutual expectations and duties.
5. For the student, such participation must not interfere in any way with the course of study and cannot be a reason for:
   (a) earning credits toward a degree
   (b) missing classes
   (c) skipping or delaying exams or any other form of evaluation
   (d) asking the university to provide extra space for desks, resources for computing, funding to attend conferences, etc

   For the accepting professor, such participation cannot be a reason for:

   (a) a reduced teaching load
   (b) extra resources or benefits of any kind (lab space, computational facilities, funds for conferences, ...)

3.11 Academic misconduct policy

3.11.1 Definitions

Primary instructor (PI) or professor – course primary instructor. **Teaching assistant (TA)** – any other instructor involved in teaching a course. **Instructor** – PI or TA.

Academic misconduct -— any action or attempted action that may result in creating an unfair academic advantage for oneself or an unfair academic advantage or disadvantage for any other member or members of the academic community. Include but not limited by:

1. cheating (including plagiarism),

2. fabrication or alteration of information and documents (lies in general),
3. theft,
4. sabotage (lesson disruption, bomb calls, fire alarms).

Students shall be considered to be **cheating** if involved in any of the following: using unauthorized cheat sheets, opening books during closed-book tests, talking during tests, plagiarising (as defined by instructor), or in any other case of cheating detected.

Plagiarism can take several forms, including but not limited to:

1. Using the exact words of another student.
2. Copying and pasting materials from the Internet or other electronic resources without proper citation, quotation or referencing.
3. Accepting excessive assistance from another person in writing.
4. Writing a computer program that is the same or closely similar to public sources or solutions of other students.

Similarity is the machine-determined score that should not exceed the given percentage. The percentage is determined by the regulation document for the exact work (e.g. course project, BS/MS thesis or coding assignment). Similarity can include quotations, in-text citations, names, titles, terms, and code. Similarity can be measured:

1. among students' works,
2. as well as with other datasets.

Similarity checks shall only be applied to long text works in natural or programming language to prove the fact of copying. For example, similarity reports are a necessary part of thesis validation procedure and coding assignments, but such reports cannot be applied to exam works and tests, where similarity does not necessarily mean copying. As there is no universal scale for similarity, for each particular assignment and course professor should clearly specify expectation for similarity in the policy document or in assignment notification. If a group of students receive the **same assignment**, not passing a similarity check shall be treated as **plagiarism**. If a student is doing a **personal task** (thesis, project, ...), high similarity cases shall be **managed by the supervisor**. In our university we use *jPlag* for code similarity detection, *Turnitin* for essay check and *Antiplagiat* for thesis check. If you want to get tutorials on these systems, please contact the Department of Education **(education@innopolis.ru)**.

3.11.2 Statement

All academic misconduct cases shall be reported to the IU Department of Education (DoE) and Student Affairs Office (SAO) using the official email address **(education@innopolis.ru, 319@innopolis.ru)**. Specify the following:

1. Student(s) name(s) involved and their roles;
2. Date and time the case was detected;
3. Course affected by the situation;
4. Short description of the situation. Please refer to this policy or other policy documents which you use to identify the misconduct.

The Student Affairs Office should create a track record of student's activities including misconduct cases starting from the bootcamp participation until the final thesis defence.

We assume that there is a publicly available cheating policy document for the whole course, particular assignment, or exam, i.e. it is clear if it is allowed or not to use electronic devices, notes, cheat sheets, books, etc. during the tests. Any violation of the said policy, as well as plagiarism cases, shall be considered equally.

The committee decision shall be taken with a consensus of all participants. The process participants shall be able to know the committee members. The committee shall be of 3 people: course professor, a teacher not involved in the course affected, and an official student representative.

Cases involving the first year students should be considered with the most thoroughness.

3.11.3 Procedure

Should an instructor (teaching assistant or professor) have reason to believe that one or more works are copied from unauthorized resources, and action on a test is taken in violation, or work of one student is copied from another (as defined in plagiarism clarification), the procedure below shall be followed.

1. The instructor shall report the case to DoE and SAO **(education@innopolis.ru and 319@innopolis.ru)**.
2. The instructor shall inform the suspected students about the above finding separately, and discuss this incident. They shall be penalized as per recommendations below and a report will be submitted to DoE outlining these decisions. In case of copying from each other — both students shall be penalized. The actions of the instructor are considered final.

3. In the event of an appeal to the DoE, the student must present a clear written rationale why the decision was in error. If the DoE agrees that the student appeal has merit then an appeal committee is formed of one student representative, the course PI, and an unrelated professor selected by the DoE. Otherwise, the decision of the course PI is considered final and the report is entered.
   (a) The student shall contact the committee and provide the said committee with the following materials: a cheating policy for this course/test/exam, students' papers/code, any other findings, including the student's arguments.
   (b) The committee shall study the above materials.
   (c) The committee can contact the students to listen to their arguments or ask questions. However, this shall be upon the committee's discretion.
   (d) The committee shall make an appeal decision to either accept or reject. This decision shall be final.
   (e) Any fabrications of data or false statements made to the DoE or the Committee during the appeals process will be seen as a separate instance of misconduct and the DoE or the Committee may impose additional outcomes if necessary when the actions have impeded investigation.

3.11.3.1 Procedure from instructor's point of view

1. Instructor reports the violation case to DoE and SAO.
2. Instructor notifies students involved (can be BCC in report email).
3. In case of student's appeal, instructor may be invited to participate in committee.

3.11.3.2 Procedure from student's point of view

1. The student gets a notification about academic misconduct.
2. The student may appeal. He/she collects the documents (3.a) and submits to DoE and SAO.

3.11.3.3 Procedure from administration point of view

1. DoE and SAO get the report about student's misconduct.
2. SAO tracks the record and asks DoE to act in case violation is not the first (see penalty section).
3. In case of student's appeal, DoE with the help of program managers collects a committee.

3.11.4 Penalty

Note that the following are the **minimal** actions.

1. Should a student be found to have committed misconduct for the first time, such a student shall receive zero points for the particular work(assignment/exam/test/homework)
2. Should a student be found to have committed misconduct for the second time, the following actions shall be taken:
    - Such a student shall:
        - receive zero points for the course (or courses if 2 cheating instances are detected in different courses);
        - be sent to retake.
    - An official warning shall be issued for the said student.
3. Should a student be found to have committed misconduct for the third time (without respect to the course), such a student shall be expelled from Innopolis University.

In the event that the actions were of a grievous nature, e.g. a student using a work for hire an assignment, a bomb/shooting threat, etc., the DoE maintains the right to increase the outcomes. A student might also be subject to additional actions on the part of the professor or DoE in order to maintain class order, have recompense of the action, or in order to learn a proper scholarly methodology, e.g. be required to make an apology to classmates affected, be required to write an essay or attend a course on how to cite properly, etc.

Part II

Structure of the BS Degrees

SUMMARY OF THE BS DEGREES

The Bachelor of Science in Computer Science and the Bachelor of Science in Computer Engineering at Innopolis University aim at creating professional software engineers, data scientists, robotics engineers, security & blockchain experts, and junior scientists who possess a deep understanding of fundamental theoretical and practical results of computer science and engineering, are able to handle the fundamental mathematical abstractions to elaborate data, can use modern tools, languages and technologies, and can understand the underlying human, social and industrial aspects of information technology.

Moreover, a person with a Bachelor of Science in Computer Science or a Bachelor of Science in Computer Engineering acquired at Innopolis University should be eminently qualified to enter the IT job market in Russia in the position of junior software developer, software engineer, robotics engineer, data scientist, or equivalent.

The primary target of these graduates is the industry of the city of Innopolis. Therefore, the program foresees many opportunities for synergies with these companies and businesses in Innopolis and the curricula reflect these industry-university synergies, leveraging substantial benefit for both parties.

Both the Bachelor of Science in Computer Science and the Bachelor of Science in Computer Engineering at Innopolis University are organised in 4 years of instruction performed mostly in English.

The first 2 years contain the fundamental courses in mathematics and physics, and in computer science and engineering. Students have the option of selecting one of two tracks: (a) *Computer Engineering*, (b) *Computer Science*.

The major goal of such an approach is to distinguish the difference in paths that consider students career goals. For instance, if a student is looking to work in cybersecurity or as a systems administrator, computer science may be a good fit for him/her. If his/her goal is to eventually become a software architect or developer, a degree in computer science or computer engineering will equip him/her for the job. Advanced computer science curricula thoroughly cover how networks and systems security protocols work while teaching programming and appropriate mathematical concepts.

Computer scientists typically have an understanding of:

- programming languages;

- how to run, maintain, and fix operating systems;
- data structures and algorithms;
- basic cybersecurity and cryptography;
- knowledge of designing, coding, and testing software;
- how computer networks work and how to manage them.

Some common skills a computer engineer utilize include:

- A complete understanding of how computer hardware and architect works;
- knowledge of designing, coding, and testing software;
- flexibility to work with a wide range of software, which can be highly specialized depending on the company and/or industry;
- ability to build your own PC systems and repair/maintain device drivers.

By the moment there is no special selection procedure for each track. Students pick the program by their own on the base of the knowledge obtained in Higher School.

In the next 2 years the students have the option of selecting one of the four streams: (a) *Software Engineering*, (b) *Data Science*, (c) *Robotics*, and (d) *Security-Blockchain*.

To provide a solid grounding for the application of the studies, there is an internship at the end of every year. This can be performed in a software company located in the city of Innopolis, or also, if such option is not practical, in a company located elsewhere in Russia, in the university labs or administration, or in another suitable institution.

The curriculum also attaches a significant importance to instruction in the humanities (history and philosophy), focusing on the aspects most relevant to ICT.

At the end of each degree students write theses, and to write theses effectively, they take a course on *Academic Research and Writing Culture*. This thesis reflects the highest standards of university education and can also be developed in collaboration with a company.

4 INTRODUCTION

4.1 Premises

The Bachelor of Science in Computer Science and Engineering at Innopolis University aims at providing its students with a quality undergraduate education in both the theoretical and applied foundations of computer science. The goal is to train students through comprehensive educational programs, and research in collaboration with industry and government, to effectively apply this education to solve real-world problems and enhance graduates' potential for high-quality lifelong careers.

The Bachelor of Science in Computer Science and Engineering at Innopolis University is organised in 4 years of instruction performed mostly in English and comprises over 240 ECTS (Europe Credit Transfer and Accumulation System).

Overall, a student takes 45 courses. The yearly course distribution (the number of core and elective courses) is the same for Software Engineering, Data Science, and Security-Blockchain streams, but it is slightly different for the Robotics stream. For more details, please see Tables 14.2 and 4.2.

Table 4.1: Distribution of the core and elective courses in Software Engineering, Data Science, and Security-Blockchain streams

Year	Core	Elective
First	12	0
Second	12	0
Third	10	1
Fourth	6	4
Total	40	5

Table 4.2: Distribution of the core and elective courses in Robotics stream

Year	Core	Elective
First	12	0
Second	12	0
Third	11	0
Fourth	8	2
Total	43	2

The internships (described later in Section 5.3) and the thesis (Appendix B) with its preparatory academic activities carry altogether 58 credits divided as specified in Table 4.3. They can be performed in a software company located in the city of Innopolis. However, if such an option is not practical, the internships and the thesis can also be performed in a company located elsewhere in Russia, within the university labs or administration, or in another suitable institution.

Table 4.3: Distribution of the credits for internships and theses

Year	Credit
First	14
Second	10
Third	10
Fourth	24
Total	58

The first 2 years (120 credits) contain the fundamental courses in mathematics and physics, and in computer science and engineering. The body of knowledge in these years consists of Math, Physics, Algorithms and Complexity, Programming Languages, Software Development Fundamentals, Architecture and Organization, Operating Systems, Computational Science, Information Management, and Networking and Communications.

In the next 2 years (120 credits) the students have the option of selecting one of the four streams: (a) *Software Engineering*, (b) *Data Science*, (c) *Robotics*, and (d) *Security-Blockchain*. The body of knowledge in these years mainly consists of Graphics and Visualization, Information Assurance and Security, Intelligent Systems, Parallel and Distributed Computing, Software Engineering, Systems Fundamentals, Human-Computer Interaction, Finance, Mechanical Engineering, Electrical Engineering, Mathematical Physics, Robotics, and Control Engineering.

The curriculum also attaches a significant importance to instruction in the humanities (history and philosophy), focusing on the aspects most relevant to ICT.

Finally, life and safety, and sport complete the educational program with 10 credits.

4.2 Organization of the information about the BS Degree

The information about the BS Degree is organized as follows. Chapter 5 on page 33 describes the overall structure of the degree in terms of distribution of credits and courses. Chapter 6 on page 36 outlines the main educational goals of the degree and the structures of the various courses of instruction. Then there are four chapters discussing the fourth streams: Chapter 7on page 47 for Software Engineering, Chapter 8 on page 52 for Data Science, Chapter 9 on page 57 for Robotics, and Chapter 10on page 62 for Security-Blockchain. Eventually, Chapter 11 on page 66 presents in depth the core courses taught in the degree and Chapter 12 on page 68 summarises the electives. In Appendix **??** on page **??** there is a graph summarising the overall curriculum and the dependencies between core courses, then in Appendix A on page 9 there is the description of the multidisciplinary exam, Appendix B on page 11 contains the procedures to elaborate and defend the thesis. Appendix C on page 22 contains the key competences acquired during the first two years of instruction while Appendix D on page 24 details the knowledge areas covered by each of the streams. Finally, Appendix E on page 28 list the main tools, technologies and programming languages introduced in the degree and Appendix F on page 30 specifies the specific courses of the direction of finance technologies.

5 STRUCTURE OF THE DEGREE

5.1 Credits

Overall, the degree requires the student to acquire 240 credits organized over 4 years of study, with a pace of 60 credits per year.

One credit of study approximately corresponds to 36 academic hours of effort per student, which can be organized differently depending on the course, as detailed further in Section 14.4. This means that a course of 4 credits requires an effort of 144 hours,[1] and an internship of 8 credits requires 288 hours (about 7 weeks).

5.2 Organization of the Courses of Instruction

Courses can be organized in a variety of ways, taking advantage of various teaching mechanisms, such as:

Lc: Class Lecture;

Tut: Class Tutorial, where the instructor explains concepts guiding students through practical exercises;

Lab: Group Laboratory, where the students perform homework and/or exercises and/or projects coordinated by the instructor, asking help from the instructor as needed, and receiving if needed also additional material directly from the instructor.

IL: Individual Labs, where the students perform exercises and refer to the instructor in person or via electronic media to get support in case of need.

During the first two years of instruction:

- Math and science courses have the structure of *2Lc-2Lab-2IL*, meaning 2 academic hours per week of class lectures, 2 academic hours of labs, and 2 academic hours of individual labs;

[1] In this document the terms “academic hour” and just “hour” are used interchangeably and mean “academic hour,” that is, 45 astronomical minutes.

- Computer science and engineering courses have the structure of *2Lc-2Tut-2Lab*, meaning 2 academic hours per week of class lectures, 2 academic hours of class tutorials, and 2 academic hours of labs.

The technical courses have the structure of *2Lc-2Lab-2IL* meaning 2 academic hours per week of class lectures, 2 academic hours of labs, and 2 academic hours of individual labs.

The courses in humanities have the structure of *2Lc-4IL*.

5.3 Internships

Every year the students have a mandatory internship during the summer break. There are three types of internships:

- Industrial – students work in Innopolis University partner IT companies, primarily located within the city of Innopolis
- Scientific – students work in Innopolis University labs on ongoing research and development tasks
- Administrative – students work in different Innopolis University departments on IT analysis, design, and implementation projects.

It is also possible to have internships that combine these three types.

The internship usually starts once the exam session for the spring semester is completed.

5.4 Knowledge of English

A graduate from the program is expected to know English at least at the level of IELTS 7. To achieve this goal, Innopolis University takes the following steps:[2]

- At admission at the university, the student must have a knowledge of English at least at level of IELTS 5.
- At the end of the first year of instruction, the student must have a knowledge of English equivalent to at least the level 5.5 of IELTS.
- At the end of the second year of instruction, the student must have a knowledge of English equivalent to at least the level 6 of IELTS.

[2] Exceptions can be provided for solidly grounded situation by the Department of Education.

- At the end of the third year of instruction, the student must have a knowledge of English equivalent to at least the level 6.5 of IELTS.
- At the end of the fouth year of instruction, the student must have a knowledge of English equivalent to at least the level 7 of IELTS.

The university will make available to the student the fundamental resources to achieve the required levels. However, it is the responsibility of the student to achieve them, taking advantage of resources provided by the university and others.

5.5 Multidisciplinary Exam

At the end of the second year of study, the student has to pass a multidisciplinary exam aimed at evaluating their comprehensive understanding of the discipline beyond the borders of an individual subject. This ensures that students who progress to years three and four have the ability to engage in system-oriented thinking and have the capacity to solve broadly-based IT problems.

The details of the multidisciplinary exam are presented in Appendix A on page 9.

5.6 Thesis

To complete successfully the study, the student has to elaborate and defend a thesis, along the lines defined in Appendix B on page 11.

5.7 Number of Admitted Students

The number of admitted students is defined by the Department of Education based on the requests of the senior university administration and of the university stakeholders.

6 Main Educational Aims and Structures

6.1 Overall Educational Goals

Innopolis University aims to prepare its students for working in various IT-related professions as software engineers, data scientists, robotics engineers, and junior scientists, to name a few. A graduate of Innopolis University should be highly qualified to enter the IT job market in Russia in the position of Junior Software Engineer or its equivalent and to be immediately productive.

A graduate of Innopolis University is able to:

1. Understand the fundamental theoretical concepts and practical results of computer science and engineering,
2. Use programming languages and other representative tools for information technology, with a firm understanding of the mathematical abstraction behind these technologies, and
3. Appreciate the underlying human, social and industrial aspects of information technology.

Innopolis University aims to create a vibrant international environment for learning. For this reason, the primary language of education at Innopolis University is English.

Innopolis University also aims to prepare its students for the constantly evolving world of information technology. A graduate of Innopolis University possesses not only IT skills but also various essential soft skills such as self-reliance, team-work, and time management.

To facilitate interaction with local and global industry, different means of delivery of the individual courses and of whole degrees are deployed, e.g., problem-based teaching, flipped classrooms, and inverted syllabi.

Innopolis University attracts top students in Russia and worldwide.

6.2 Organization of Studies

The curriculum spans four years.

First and second-year students can choose one of the following tracks:

- Computer Engineering
- Computer Science

Despite their choice students take fundamental courses in Engineering, Mathematics and Computer Science in each semester but in different depths of studying.

Each track contains the same set of core courses, but the courses that are taught in different level of depth have different titles, e.g. Calculus-I vs Mathematical Analysis-I, Analytical Geometry - I vs Essentials of Analytical Geometry-I, Programming Software Systems-I vs Introduction to Programming-I.

Also, these tracks have a set of courses with lectures that are taught for all the students simultaneously but with different approach to practical classes, e.g. Computer Architecture (Fundamentals of Computer Architecture) and Discrete Math, Phylosophy(Logic).

Computer science track focuses on topics in computational theory. These include the virtual aspects of computers, focusing on software, rather than hardware. As a field that is closely aligned with mathematics, computer science applies theoretical ideas to solve real world problems. Computer science degree programs require courses including analysis of algorithms, operating system principles, computer architecture and software engineering, so an interest in math, puzzles, and problem solving would suit a student well. A degree in computer science will cover essential hardware and software topics, including computer organization and architecture.

Computer science is often described as more abstract and less hands-on than computer engineering. As a computer scientist, students will focus on using computational theory, mathematics and data structures to write effective codes.

Computer engineering track focuses on how to build devices including robots. It is a field that combines physics, electrical engineering and computer science. The focus of computer engineering is on hardware, rather than software and it's closer to Robotics rather than to Software Development.

The work of a computer engineer works in the physical world and involves understanding how we can harness the laws of physics and electronics to create better computer components. They are more likely to spend more time at a lab bench than writing code. From this point of view computer engineering is often described as more practical activity than computer engineering. Learning the same subjects students will be more focused on the applications of the theory for solving real life problems.

Third and fourth-year students can choose one of the following streams:

- Software engineering

- Data science
- Robotics
- Security-Blockchain

Each stream has its own set of specific courses, as well as those that are shared with other streams. Moreover, the stream in data science has a direction in finance technologies, specializing data science for the banking and financial industry.

The organization of the courses is as detailed in Tables 14.2 and 4.2. In addition to them there are courses for sport in every semester. Furthermore, students of each stream do a project in the spring semester of the third year, and a thesis in both semesters of the final (fourth) year.

Each of the taught courses (core and electives) and the thesis course done during the semester count for 4 credits. The sport courses count for one credit each. The Life Safety course counts for 2 credits. The summer internship counts for 14 credits in the first year, 10 credits in the second and third years, and 8 credits in the fourth year.

Table **??** shows an overview of the curriculum for the complete BS program. The details of the core courses are in Chapter 11 on page 66, and those of the electives are in Chapter 12 on page 68.

Table 6.1: First and second year of the Bachelor Curriculum.

Semester	BSc in Computer Science	BSc in Computer Engineering
BSc-1-Fall	Introduction to Programming I	Programming Software Systems I
	Computer Architecture	Computer Architecture
	Mathematical Analysis I	Calculus I
	Analytic Geometry and Linear Algebra I	Fundamentals of Analytic Geometry and Linear Algebra I
	Discrete Mathematics	Discrete Mathematics
	Philosophy (Logic)	Fundamentals of Philosophy (Logic)
	English	English
	Sport	Sport
BSc-1-Spring	Introduction to Programming II	Programming Software Systems II
	Data Structures and Algorithms	Data Structures and Algorithms
	Mathematical Analysis II	Calculus II
	Analytic Geometry and Linear Algebra II	Fundamentals of Analytic Geometry and Linear Algebra II
	Theoretical Computer Science	Theoretical Computer Science
	English	English
BSc-1-Summer	Internship I	
BSc-2-Fall	Probability and Statistics	
	Physics I (Mechanics)	
	Introduction to Software Engineering	
	Operating Systems	
	Differential Equations	
	History	
BSc-2-Spring	Physics II (Electromagnetism)	
	Networks	
	Control Theory	
	Artificial Intelligence	
	Database Systems	
	Elective	
BSc-2-Summer	Internship II	

Table 6.2: Third and forth year of the Bachelor Curriculum.

Semester	Software Engineering (SE)	Data Science (DS)	Robotics (R)	Security-Blockchain (SB)
BSc-3-Fall	Information Theory			
	Introduction to Machine Learning			
	Philosophy II (Languages and Perceptions)			
	Compilers Construction	Introduction to Big Data	Non-linear Optimization	Fundamentals of Computer Security
	Software Architecture	Software Architecture	Theoretical Mechanics	System and Network Administration
	Distributed Systems	Distributed Systems	Fundamentals of Robotics	Distributed Systems
	Elective			
BSc-3-Spring	Digital Signal Processing			
	Game Theory	Game Theory	Robotic Systems	Game Theory
	Lean Software Development	Information Retrieval	Mechanics and Machines	Information Retrieval
	Software System Design	Data Mining	Sensors and Sensing	Network and Cyber Security
	Tech Elective 1 [a]			
	Tech Elective 2			
BSc-3-Summer	Internship III			
BSc-4-Fall	Academic Research and Writing Culture I			
	Thesis I			
	Thesis II			
	Mobile App Development	Numerical Modeling	Numerical Modeling	Distributed Ledger Technologies
	Economics	Introduction to Computer Vision	Introduction to Computer Vision	Economics
	Elective III	Elective III	Mobile Robotics and Autonomous Driving	Elective III
	Elective IV	Elective IV	Contact-aware control	Elective IV
BSc-4-Spring	Academic Research and Writing Culture II			
	Thesis III			
	Thesis IV			
	Life safety			
	Software Quality	Practical Machine Learning and Deep Learning	Practical Machine Learning and Deep Learning	Business Analytics
	Microcontrollers	Statistical Techniques for Data Science	Microcontrollers	Microcontrollers
	Elective	Elective	-	Elective
BSc-4-Summer	Internship IV			

[a] A project in DS/NB/R/SE, running throughout the semester

[b] A course on how to write a SLR, running throughout the semester, called Technical Literature Review.

6.3 Main Areas and Knowledge Areas

To effectively design the curriculum, we identify areas that must be covered in the curriculum. We distinguish the following two kinds of areas, i.e., main areas and knowledge areas. First, the main areas are the key components of the degree program listed as follows:

- (CSE) Computer Science and Engineering
- (M) Math
- (P) Physics
- (H) Humanities
- (M-CS) Math and Computer Science
- (IT) Internship and Theses

Meanwhile, the knowledge areas shown in the following list identify more specifically the body of knowledge that must be covered in the curriculum.

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

- (AO) Architecture and Organization
- (SP) Social Issues and Professional Practice

We identified the knowledge areas based on the Curriculum Guidelines for Undergraduate Degree Programs in Computer Science by the joint task force between ACM (Association for Computing Machinery) and IEEE Computer Society in 2013.[1] Note that a knowledge area is a higher level concept than a course, and an individual course can cover diverse knowledge areas when appropriate.

We provide throughout the document how each course is mapped to a main area and knowledge areas. We also provide the distribution of the knowledge areas in the first two years, and in the three specialised streams.

6.4 Structure of the First Two Years of Instruction

The primary objective of the first two-year curriculum is to provide a solid scientific and technical foundation for a career in Computer Science and Engineering, prior to the development of specialized knowledge and skills, which occurs in the second two years of the curriculum.

More especially, the aim of the first two years is to foster the following fundamental skills and knowledge:

- Ability to develop computer programs and systems efficiently. Related courses are *Introduction to Programming* and *Software Project* where the former focuses on improving the programming skills of students, while the latter also involves other diverse software development activities such as software design and maintenance.
- Ability to solve problems efficiently using computer programs. Related courses are *Data Structures and Algorithms*, and *Introduction to AI*.
- Ability to understand the theory and concept of computer systems. Related courses are *Computer Architecture, Data Modeling and Databases, Operating Systems, Networks,* and *Theoretical Computer Science.*

To this end, the curriculum emphasizes building mathematical foundations and covers diverse math subjects—*Mathematical Analysis, Differential Equations, Analytic Geometry and Linear Algebra, Discrete Math*, and *Probability and Statistics.* With strong math skills,

[1]Association for Computing Machinery (ACM) Joint Task Force on Computing Curricula and IEEE Computer Society. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. ACM, New York, NY, USA, 2013. ISBN 978-1-4503-2309-3. 999133

Table 6.3: First Year of the BS in Computer Science and Engineering

Course Name	Main Area	Knowledge Area	Teaching Structure
Fall Semester			
Introduction to Programming I	CSE	PL, SE	2Lc-2Tut-2Lab
Computer Architecture	CSE	AO	2Lc-2Tut-2Lab
Discrete Math	M	MH, CN	2Lc-2Lab-2IL
Mathematical Analysis I	M	MH, CN	2Lc-2Lab-2IL
Analytic Geometry and Linear Algebra I	M	MH, CN	2Lc-2Lab-2IL
Philosophy (Logic)	M	MH, CN	2Lc-2Lab-2IL
Spring Semester			
Introduction to Programming II	CSE	PL, SE	2Lc-2Tut-2Lab
Data Structures and Algorithms	CSE	AL, PL, SDF	2Lc-2Tut-2Lab
Mathematical Analysis II	M	MH, CN	2Lc-2Lab-2IL
Analytic Geometry and Linear Algebra II	M	MH, CN	2Lc-2Lab-2IL
Theoretical Computer Science	CSE	MH, PL, SDF	2Lc-2Tut-2Lab
Summer Semester			
Internship I	IT		6IL

Table 6.4: Second Year of the BS in Computer Science and Engineering

Course Name	Main Area	Knowledge Area	Teaching Structure
Fall Semester			
Probability and Statistics	M	MH, CN	2Lc-2Lab-2IL
Physics I *(Mechanics)*	P	PH	2Lc-2Tut-2Lab
Operating Systems	CSE	OS, PD	2Lc-2Tut-2Lab
Fundamentals of Software Engineering	CSE	AO, SDF, SE	2Lc-2Lab-2IL
Differential Equations	M	MH	2Lc-2Lab-2IL
History	H	Hs	2Lc-2Lab-2IL
Spring Semester			
Physics II (Electrical and Electronic Circuits)	P	PH, EE	2Lc-2Tut-2Lab
Networks	CSE	CM	2Lc-2Tut-2Lab
Control Theory (Linear Control)	M	CE	2Lc-2Tut-2Lab
Artificial Intelligence	CSE	AL	2Lc-2Tut-2Lab
Databases	CSE	SDF, SE	2Lc-2Lab-2IL
Software Project	CSE	SE	2Lc-2Tut-2Lab
Summer Semester			
Internship II	IT		6IL

students will be able to grasp any advanced subjects in their interest domain of information technology.

The curriculum also provides basic subjects on Physics and Electronics. Knowledge of these subjects will be essential in pursuing further study in Robotics, and other areas of information technology such as Cyber-Physical Systems, and Internet of Things.

Specifically, the students during the first year (Table 6.3) take courses on the most fundamental computer science/engineering subjects (*Introduction to Programming* I/II, *Data Structures and Algorithms*, *Computer Architecture*), math (*Mathematical Analysis* I/II, *Analytic Geometry and Linear Algebra* I/II, *Discrete Math*, *Probability and Statistics*), and physics (*Physics* I). They are the languages of information technology, and will be the fundamental foundation when students pursue their specialized streams in their last two years of the program.

During the second year (Table 6.4), students take essential computer science/engineering courses such as *Data Modeling and Databases* I/II, *Operating Systems*, *Software Project*, *Networks*, and *Theoretical Computer Science*. Also, *Introduction to AI* is, given its importance, taken by all students. As in the first year, students also take courses on math (*Differential Equations*, *Stochastic Processes*). Lastly, students also take a fundamental course on electronics (*Physics* II) in the second year.

Last but not least, internship (*Internship* I/II) is a core part of the curriculum throughout the first two years. Students should complete their internship every summer.

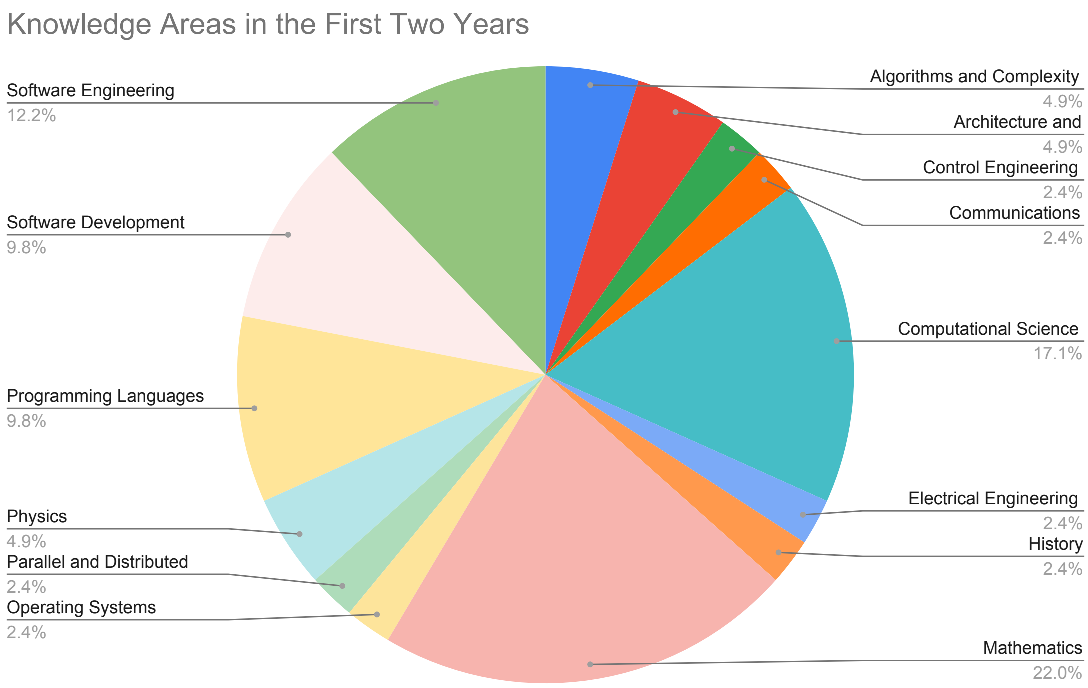

Figure 6.1: Breakdown of the knowledge areas (%) in the courses of the first years.

After the first two years of studies, the student will have a general understanding of the general key competences discussed later in Appendix **??**. Percentage-wise summary of the courses from the first and the second year covering different knowledge areas is given in Table 6.5.

6.5 Overview of the Second Two Years of Instruction

Equipped with fundamental competencies in math, engineering, and computer science in the first two years of the BS program, the students enter the third and the fourth year

Table 6.5: Mapping between knowledge areas and courses in the first two years

Knowledge	Courses	Portion
Math	*Analytic Geometry and Linear Algebra* I/II *Mathematical Analysis* I/II *Discrete Math* *Philosophy(Logic)* *Differential Equations* *Probability and Statistics* *Theoretical Computer Science*	22 %
Computational Science	*Analytic Geometry and Linear Algebra* I/II *Mathematical Analysis* I/II *Discrete Math* *Philosophy (Logic)* *Probability and Statistics*	17 %
Software Engineering	*Introduction to Programming* I/II *Fundamentals of Software Engineering* *Databases* *Software Project*	12 %
Software Development Fundamentals	*Data Structures and Algorithms* *Databases* *Fundamentals of Software Engineering* *Theoretical Computer Science*	10 %
Programming Languages	*Data Structures and Algorithms* *Introduction to Programming* I/II *Theoretical Computer Science*	10 %
Algorithms and Complexity	*Data Structures and Algorithms* *Artificial Intelligence*	5 %
Architecture and Organization	*Computer Architecture* *Fundamentals of Software Engineering*	5 %
Physics	*Physics* I (Mechanics) *Physics* II (Electrical and Electronic Circuits)	5 %
Control Engineering	*Control Theory*	2 %
Communications	*Networks*	2 %
Electrical Engineering	*Physics* II (Electrical and Electronic Circuits)	2 %
History	*History*	2 %
Operating Systems	*Operating Systems*	2 %
Parallel and Distributed Programming	*Operating Systems*	2 %

where they are trained on four different career paths having the fastest growing employment opportunities in the world. The four streams are:

- Software engineering,
- Data science,
- Robotics,
- Security-Blockchain

The Software Engineering Stream combines system theory, computer science, and software engineering for students interested in pursuing careers in software engineering. The

Data Science Stream integrates intelligent systems, computer science, and data science to produce graduates with the skills needed to evaluate and interpret big data. The Robotics Stream is aimed at giving fundamental knowledge in mechanics, mechatronics, electrical engineering, control theory, and robotics to provide students an understanding of the basic principles of robotic systems in particular and technical systems in general. Finally, the Security-Blockchain stream focuses on information security, relating to the design and development of secure computer software, providing in-depth handling of the current landscape of vulnerabilities, risks and security disciplines. Alongside computer security, the stream also covers aspects related to distributed leadgers technologies (e.g., cryptocurrencies and blockchain technologies).

Students take core courses through the third and fourth year, some of which belong to individual streams while others are shared. They also take one-five electives, one in the second semester of the third year and others in the fourth year. Students also need to complete a project in the third year. During the fourth year, students need to complete their theses supervised by Innopolis University faculty.

7 SOFTWARE ENGINEERING STREAM

7.1 Specific Educational Goals of the Program

The Software Engineering Stream is designed for students interested in pursuing careers in software engineering. The Software Engineering Stream mainly consists of courses that provide students with a core background in designing, building, validating and maintaining complex software systems. As machine learning is becoming one of the core competences required for software engineers, courses relevant to that topic are also provided to students in the Software Engineering Stream.

Learning Outcomes: A student completing the Software Engineering Stream will be able to:

- Solve difficult computer systems problems in a creative and innovative way.
- Design a system, or a component of it, in order to meet the desired needs within realistic economic and environmental constraints.
- Understand the importance of componentization and reuse.
- Use models, techniques and technologies available to cover all the phases of software development, from requirements elicitation to design, implementation and testing.
- Apply established and innovative service-based software architecture paradigms and design patterns to software development.
- Take advantage of the most recent software verification techniques and tools, such as model checking and various static analysis techniques.
- Apply the principles of syntactic and semantic analysis in compilers.
- Distinguish among the various existing models for software processes, both traditional and agile, and be able to identify, customize, and apply the most suitable depending on circumstances.
- Work both individually and as a team member in order to develop and deliver software artifacts of high quality.
- Develop communication skills, negotiation skills, work ethic and discipline, and leadership.
- Learn how to support and advance professional and organizational goals.

7.2 Key Competences and Practices

The graduates of the Software Engineering Stream will have the following competencies after successful completion of the course of studies.

1. Software Development Methodologies
2. Software Design Methodologies and Tools
3. Requirements Engineering
4. Systems Specification Techniques
5. Testing Methodologies
6. Software Architecture
7. Design Patterns
8. Compilers Construction
9. Static Analysis
10. Model Checking
11. Concurrency Theory
12. Design-by Contract
13. Agile Methods
14. Programming Paradigms
15. Component-Based Development
16. Service-Based Systems
17. Software Metrics
18. Programming Relational and NoSQL DB
19. User Interface Design
20. Code Quality
21. Analysis
22. Presentation
23. Communication
24. Technical Writing
25. Technical Reading
26. Problem Solving
27. Focus on User or Customer Needs
28. Cost/Benefit Analysis
29. Teamwork
30. Leadership

7.3 Structure of the Two Years of Specialization

During the third and fourth year of this stream, the students take a set of core courses for the Software Engineering Stream.[1] Tables 7.1, 7.2, 7.3, and 7.4 show the list of courses for the Software Engineering Stream. These tables also show the main areas and the knowledge areas covered by each course, along with its teaching structure.

Students also take electives course in these years, starting from the Spring semester of the third year. While students can choose any electives they want from various available courses, at least one of the three electives must be about a technical subject, and another about soft-skill, humanities, or economics/marketing/finance. The details of the electives are presented in Chapter 12.

In addition, one of the elective that students can choose is a Software Engineering project

[1] Some of the courses are shared with other streams.

Table 7.1: Third Year: SE Stream (Fall Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Information Theory	CSE	IS, GV	2Lc-2Lab-2IL
Software Architecture	CSE	SE	2Lc-2Lab-2IL
Compilers Construction	CSE	PL, AL	2Lc-2Lab-2IL
Introduction to Machine Learning	CSE	IS	2Lc-4IL
Distributed Systems & Cloud Computing	CSE	PD	2Lc-2Lab-2IL
Philosophy II *(Languages and Perceptions)*	H	Ph	2Lc-4IL

Table 7.2: Third Year: SE Stream (Spring Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Software Systems Design	CSE	SE	2Lc-2Lab-2IL
Digital Signal Processing	CSE	CM, EE	2Lc-2Lab-2IL
Game Theory	CSE	IS, CN, AL	2Lc-2Lab-2IL
Lean Software Development	CSE	SE, SDF	2Lc-2Lab-2IL
Elective I			
Elective II			
Elective III			

Table 7.3: Fourth Year: SE Stream (Fall Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Microcontrollers & Embedded Hardware	CSE	OS, PL	2Lc-2Lab-2IL
Mobile Application Development	CSE	PL	2Lc-2Lab-2IL
Economics & Finance	CSE	SP	2Lc-2Lab-2IL
Elective IV			
Academic Research and Writing Culture	H	SP	
Thesis	IT		

Table 7.4: Fourth Year: SE Stream (Spring Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Software Quality & Reliability	CSE	SE	2Lc-2Lab-2IL
Elective V			
Elective VI			
Thesis	IT		
Life Safety	H	SP	

supervised by Innopolis University faculty or Industry, again in the Spring semester of the third year. Finally, students must complete thesis work in the fourth year.

Summarized information about the Software Engineering Stream is shown in Figure 7.1 and Table 7.5. Figure 7.1 shows the breakdown of the knowledge areas covered in the Software Engineering Stream, and Table 7.5 shows the courses covering each knowledge area.

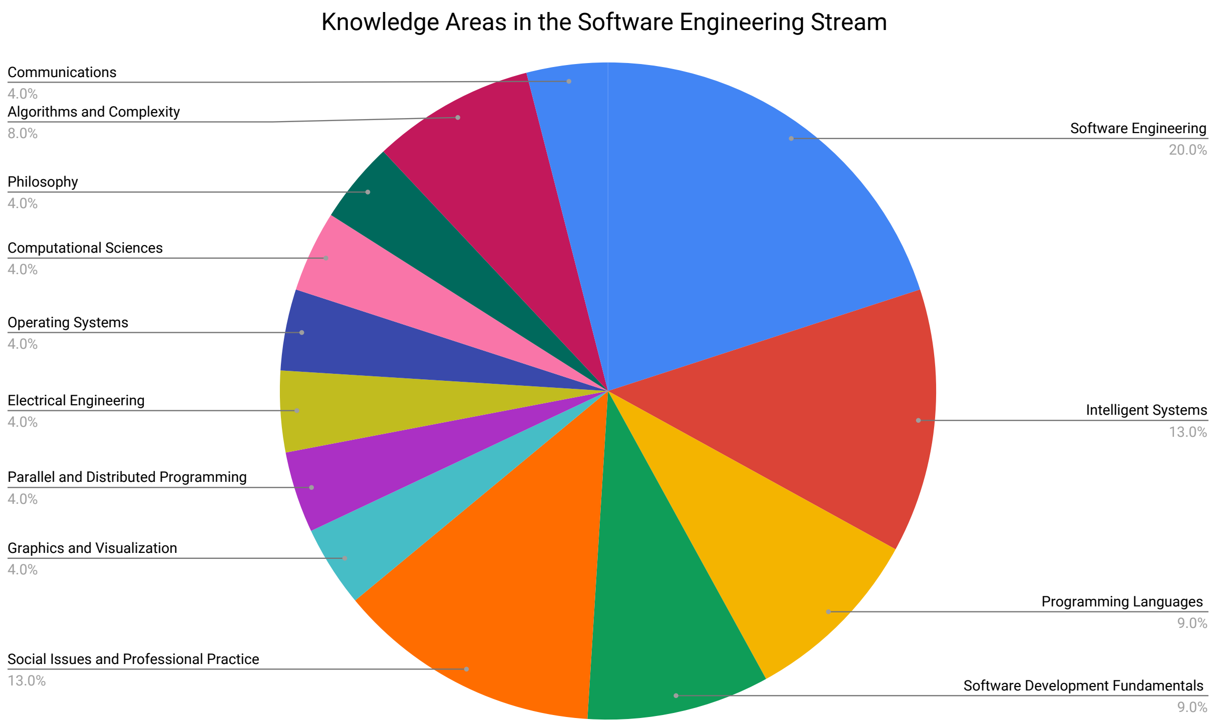

Figure 7.1: Breakdown of the knowledge areas (%) in the Software Engineering Stream.

Table 7.5: Mapping between knowledge areas and courses in the Software Engineering Stream

Knowledge	Courses	Portion
Software Engineering	*Software Systems Design* *Lean Software Development* *Software Architectures* *Software Quality & Reliability*	20 %
Intelligent Systems	*Information Theory* *Introduction to Machine Learning* *Game Theory*	13 %
Programming Languages	*Compilers Construction* *Microcontrollers & Embedded Hardware*	9 %
Software Development Fundamentals	*Lean Software Development* *Mobile Application Development*	9 %
Social Issues and Professional Practice	*Academic Research and Writing Culture* *Life Safety* *Economics & Finance*	13 %
Graphics and Visualization	*Information Theory*	4 %
Parallel and Distributed Programming	*Distributed Systems & Cloud Computing*	4 %
Electrical Engineering	*Digital Signal Processing*	4 %
Operating Systems	*Microcontrollers & Embedded Hardware*	4 %
Computational Sciences	*Game Theory*	4 %
Philosophy	*Philosophy* 2 (Languages and Perceptions)	4 %
Algorithms and Complexity	*Compilers Construction* *Game Theory*	8 %
Communications	*Digital Signal Processing*	4 %

8 Data Science Stream

8.1 Specific Educational Goals of the Program

The Data Science Stream is designed for students interested in pursuing careers in data science and analysis. The Data Science Stream consists of courses that provide students with a core background in algorithmic, statistical, and systems expertise with a focus on acquiring, storing, accessing, analyzing and visualizing large, heterogeneous and real-time data. These data are generally associated with various real-world domains including health, media, energy, and transportation. Students will have a chance to learn the latest ideas and emerging techniques in the field as well as to gain experience with industrial projects.

Learning Outcomes: A student completing the Data Science Stream will be able to:

- Manage and analyze a large volume of data in a scalable fashion.
- Pick the right tools to manage and analyze a large volume of data.
- Develop a data analytic tool on an appropriate platform, including a cloud computing platform.
- Understand in detail how machine learning works.
- Apply an appropriate machine learning technique to a real-world problem at hand.
- Develop a data mining system appropriate for the task at hand.
- Understand in detail how an information retrieval system works.
- Develop an information retrieval system appropriate for the task at hand.
- Collect a large volume of data necessary for data analytics.
- Understand various database systems, and pick an appropriate database system for the task at hand.
- Understand how decision-makers use business analytics to formulate and solve business problems.
- Understand and apply the appropriate tools for the analysis of quantitative and qualitative data.
- Use software packages fluently for data analysis.
- Solve challenging problems in a creative way.

8.2 Key Competences and Practices

The graduates of the Data Science Stream will have the following competencies after successful completion of the course of studies.

1. Distributed Databases
2. Relational Databases
3. Big Data
4. Database Design
5. Database Systems
6. Data Mining
7. Data Modeling
8. Machine Learning
9. Deep Learning
10. Data, Information, and Knowledge
11. Probability
12. Programming
13. Query Languages
14. Code Quality
15. Analysis
16. Presentation
17. Communication
18. Technical Writing
19. Technical Reading
20. Problem Solving
21. Teamwork
22. Leadership

8.3 Structure of the Two Years of Specialization

During the third and fourth year of this stream, the students take a set of core courses (some of which shared with other streams). They also take electives starting from the second semester (spring semester) of the third year. Details on the electives are presented in Chapter 12. At least one of these electives needs to be technical and one related to soft skills, humanities, or economics/marketing/finance.

Note that the students can also choose as an elective to complete a data science project supervised by Innopolis University faculty in the second semester of the third year. Students must do their thesis work in the fourth year. Summarized information containing the main areas, the knowledge areas and the teaching structure for each course is presented in tables 8.1, 8.2, 8.3 and 8.4. Furthermore, percentage-wise summary of the courses in the Data Science Stream covering different knowledge areas is given in Table 8.5, whereas the breakdown of the knowledge areas across two years is presented in Figure 8.1.

Table 8.1: Third Year: DS Stream (Fall Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Information Theory	CSE	IS, GV	2Lc-2Lab-2IL
Software Architecture	CSE	SE	2Lc-2Lab-2IL
Introduction to Big Data	CSE	IS, AL	2Lc-2Lab-2IL
Introduction to Machine Learning	CSE	IS	2Lc-4IL
Distributed Systems	CSE	PD	2Lc-2Lab-2IL
Philosophy II *(Languages and Perceptions)*	H	Ph	2Lc-4IL
Hum. Elective I			

Table 8.2: Third Year: DS Stream (Spring Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Information Retrieval	CSE	IS	2Lc-2Lab-2IL
Digital Signal Processing	CSE	CM, EE	2Lc-2Lab-2IL
Game Theory	CSE	IS, CN, AL	2Lc-2Lab-2IL
Data Mining	CSE	IS	2Lc-2Lab-2IL
Elective I			
Elective II			

Table 8.3: Fourth Year: DS Stream (Fall Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Numerical Modeling	M-CS	CN, MP	2Lc-2Lab-2IL
Introduction to Computer Vision	CSE	IS	2Lc-2Lab-2IL
Academic Research and Writing Culture	H	SP	
Thesis I	IT		
Thesis II	IT		
Elective III			
Elective IV			

Table 8.4: Fourth Year: DS Stream (Spring Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Practical Machine Learning & Deep Learning	CSE	IS	2Lc-2Lab-2IL
Statistical Techniques for Data Sceicne	M-CS	CN	2Lc-2Lab-2IL
Thesis III	IT		
Thesis IV	IT		
Life Safety	H	SP	
Hum. Elective II			

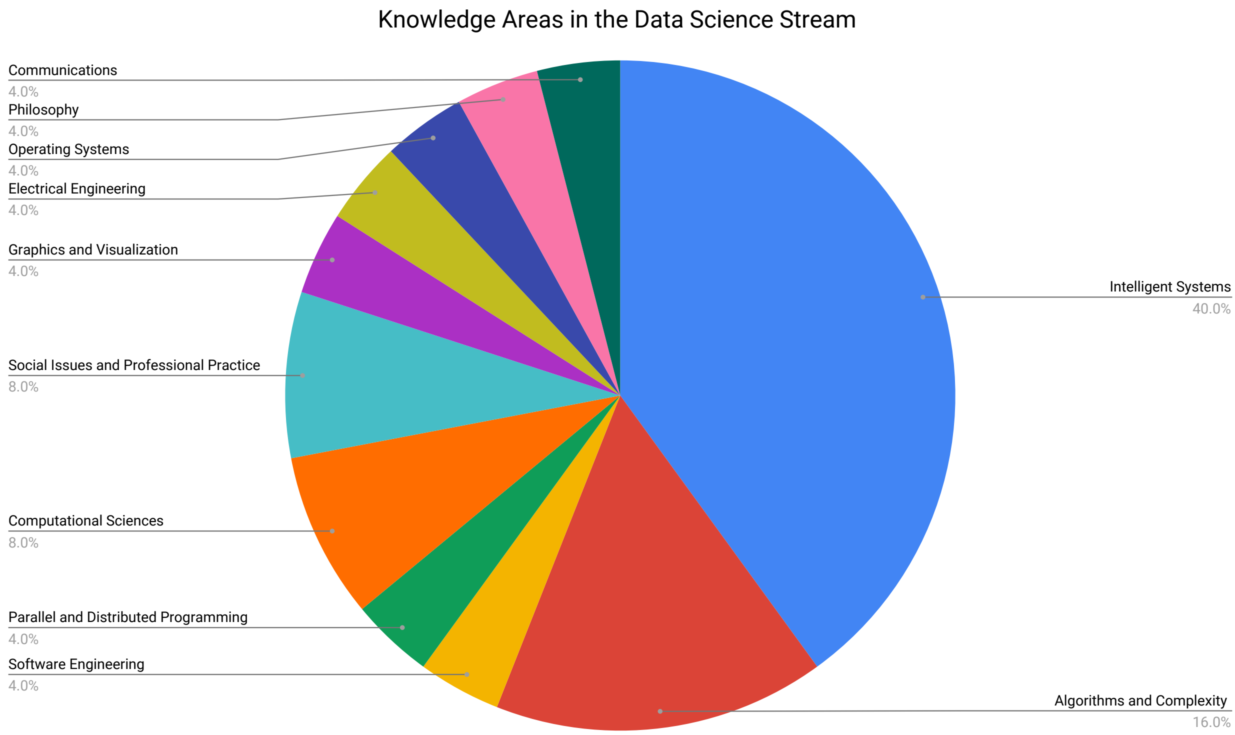

Figure 8.1: Breakdown of the knowledge areas (%) in the Data Science Stream.

Table 8.5: Mapping between knowledge areas and courses in the DS Stream

Knowledge	Courses	Portion
Intelligent Systems	*Information Theory* *Introduction to Big Data* *Introduction to Machine Learning* *Information Retrieval* *Data Mining* *Practical Machine Learning & Data Mining* *Game Theory* *Introduction to Computer Vision*	40 %
Algorithms and Complexity	*Introduction to Big Data* *Game Theory* *Numerical Modeling*	16 %
Software Engineering	*Software Architecture*	4 %
Parallel and Distributed Programming	*Distributed Systems*	4 %
Computational Science	*Statistical Techniques for Data Science* *Game Theory* *Numerical Modeling*	8 %
Social Issues and Professional Practice	*Academic Research and Writing Culture* *Life Safety*	8 %
Graphics and Visualization	*Information Theory*	4 %
Electrical Engineering Engineering	*Digital Signal Processing*	4 %
Operating Systems	*Embedded and Shared Memory Programming*	4 %
Philosophy	Philosophy II *(Languages and Perceptions)*	4 %
Communications	*Digital Signal Processing*	4 %

9 Robotics Stream

9.1 Specific Educational Goals of the Program

The primary goal of the Robotics Stream is to provide fundamental knowledge in mechanics, mechatronics, electrical engineering, control theory and robotics that will allow students to understand the basic principles of robotic systems in particular and technical systems in general. Students will have the opportunity to learn the latest trends and techniques emerging in the field and practice them in industrial-relevant projects. The obtained knowledge and skills will be in demand not only in robotics companies, but also in any engineering tasks, project works, R&D centers and others. The graduates will gain the necessary knowledge to create a new product and start their own business.

Learning Outcomes: A student completing the Robotics Stream will be able to:

- Model the kinematics of robotic systems.
- Formulate and compute dynamic equations necessary to control a robotic system.
- Write code for sensor data processing.
- Understand the physical principles of sensors and their limitations.
- Design, implement, and document appropriate, effective, and efficient software solutions for a variety of real-world computer vision problems.
- Apply their knowledge of image acquisition, image processing, and image analysis to extract useful information from visual images.
- Build virtual reality applications.
- Build real-time embedded systems.
- Understand the characteristics of control systems, including controllability, stability and regulation quality.
- Perform time and frequency analysis for control systems including nonlinear control systems.
- Solve diverse mechanics problems using various methods such as Newtonian, Lagrangian and Hamiltonian methods.

9.2 Key Competences and Practices

The graduates of the Robotics Stream will have the following competencies after successful completion of the course of studies.

1. Robotics
2. Control Programs
3. Control system analysis
4. Mechanical Systems
5. Mathematical Modeling
6. Sensing
7. Perception and Computer Vision
8. Time and Frequency Analysis
9. Optimization
10. Numerical Methods
11. Artificial Intelligence
12. Mechatronics
13. Machine Learning
14. ~~Deep Learning~~
15. ~~Computer Graphics~~
16. ~~Virtual Realities~~
17. ~~Matlab~~ Octave, SciLab
18. ~~Simulink~~
19. ~~Real Time and Embedded Systems~~
20. ~~Assembly Language~~
21. ~~Digital Logic and Digital Systems~~
22. ~~Analysis~~
23. ~~Presentation~~
24. ~~Communication~~
25. Technical Writing
26. Technical Reading
27. ~~Problem Solving~~
28. ~~Cost/Benefit Analysis~~
29. ~~Teamwork~~
30. ~~Leadership~~

9.3 Structure of the Two Years of Specialization

During the third and fourth year of this stream, the students take a set of core courses (some of which are shared with other streams) and also three electives, one in the second semester of the third year and two in the fourth year, one each semester (details on the electives are presented in Chapter 12), at least one of which needs to be technical and one related to soft skills, humanities, or economics/marketing/finance.

Note that the students can also choose as an elective to complete a robotics project supervised by Innopolis University faculty in the second semester of the third year. Students must do their thesis work in the fourth year. Summarized information containing the main areas, the knowledge areas and the teaching structure for each course is presented in tables 9.1, 9.2, 9.3 and 9.4. Furthermore, percentage-wise summary of the courses in the Robotics Stream covering different knowledge areas is given in Table 9.5, whereas the breakdown of the knowledge areas across two years is presented in Figure 9.1.

Table 9.1: Third Year: Robotics Stream (Fall Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Information Theory	CSE	IS, GV	2Lc-2Lab-2IL
Control Theory	CSE	CE	2Lc-2Lab-2IL
Introduction to Robotics	CSE	IS	2Lc-2Lab-2IL
Introduction to Machine Learning	CSE	IS	2Lc-4IL
Theoretical Mechanics	M	MP	2Lc-2Lab-2IL
Philosophy II *(Languages and Perceptions)*	H	Ph	2Lc-4IL

Table 9.2: Third Year: Robotics Stream (Spring Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Non-linear Optimization	CSE	IS	2Lc-2Lab-2IL
Digital Signal Processing	CSE	CM	2Lc-2Lab-2IL
Robotic Systems	CSE	IS	2Lc-2Lab-2IL
Sensors & Sensing	CSE	IS	2Lc-4IL
Mechanics & Machines	M	MP	2Lc-2Lab-2IL
Elective I			
Elective II			

Table 9.3: Fourth Year: Robotics Stream (Fall Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Microcontrollers & Embedded Hardware	CSE	OS, PL, EE	2Lc-2Lab-2IL
Numerical Modeling	M-CS	CN, MP	2Lc-2Lab-2IL
Introduction to Computer Vision	CSE	IS	2Lc-2Lab-2IL
Elective III			
Academic Research and Writing Culture	H	SP	
Thesis	IT		

Table 9.4: Fourth Year: Robotics Stream (Spring Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Practical Machine Learning & Deep Learning	CSE	IS	2Lc-2Lab-2IL
Mechatronics	CSE	EE, ME	2Lc-2Lab-2IL
Mobile Robotics & Autonomous Driving	CSE	IS	2Lc-2Lab-2IL
Thesis	IT		
Life Safety	H	SP	

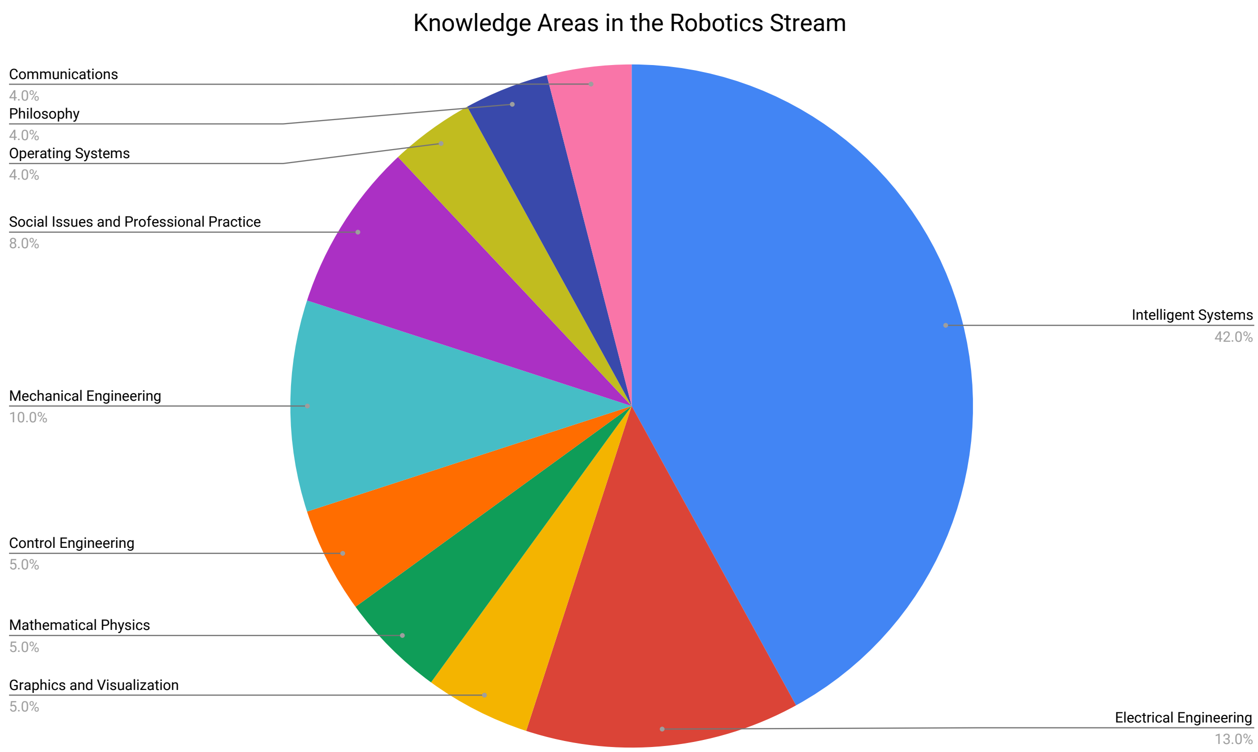

Figure 9.1: Breakdown of the knowledge areas (%) in the Robotics Stream.

Table 9.5: Mapping between knowledge areas and courses in the Robotics Stream

Knowledge	Courses	Portion
Intelligent Systems	*Information Theory* *Introduction to Robotics* *Introduction to Machine Learning* *Introduction to Computer Vision* *Sensors and Sensing* *Advanced Robotics* *Mobile Robotics & Automonous Driving* *Robotic Systems* *Practical Machine Learning & Deep Learning*	42 %
Electrical Engineering	*Mechatronics* *Microcontrollers and Embedded Hardware* *Digital Signal Processing*	13 %
Graphics and Visualization	*Information Theory*	4 %
Mathematical Physics	*Mechanics & Machines* *Numerical Modeling*	5 %
Control Engineering	*Non-Linear Optimization*	5 %
Mechanical Engineering	*Mechatronics* *Mechanics and Machines*	10 %
Social Issues and Professional Practice	*Academic Research and Writing Culture* *Life Safety*	8 %
Operating Systems	*Microcontrollers and Embedded Hardware*	4 %
Philosophy	Philosophy 2 (Languages and Perceptions)	4 %
Communications	*Digital Signal Processing*	4 %

10 SECURITY-BLOCKCHAIN STREAM

10.1 Specific Educational Goals of the Program

The Security-Blockchain stream is designed for students interested in pursuing careers in information security and blockchain. It mainly consists of courses that provide students with a core background in designing, and building secure software systems. The stream also covers distributed ledgers technologies (e.g., cryptocurriencies and blockchain technologies).

Learning Outcomes: A student completing the Security-Blockchain stream will be able to:

- Develop robust and secure code.
- Reverse engineer software into formats that can be deciphered and analyzed.
- Explore cyber security and information assurance concepts for secrecy, integrity, and availability of information.
- Develop a deeper understanding of blockchain technology and its long-term implications for business, coupled with knowledge of its relationship to other emerging technologies such as AI.
- Make better strategic business decisions by drawing on their knowledge of blockchain and affiliated industries and technologies.
- Work both individually and as a team member in order to develop and deliver secure software artifacts of high quality.
- Develop communication skills, negotiation skills, work ethic and discipline, and leadership.
- Learn how to support and advance professional and organizational goals.

10.2 Key Competences and Practices

The graduates of the Security-Blockchain will have the following competencies after successful completion of the course of studies.

1. Information Security Assurance
2. Security Policies and Laws
3. Software Security
4. Network security
5. Distributed ledgers
6. Finance

Table 10.1: Third Year: SB Stream (Fall Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Information Theory	CSE	IS, GV	2Lc-2Lab-2IL
Fundamentals of Computer Security	CSE	IAS, SP	2Lc-2Lab-2IL
System and network administration	CSE	IAS, SP	2Lc-2Lab-2IL
Introduction to Machine Learning	CSE	IS	2Lc-4IL
Distributed Systems	CSE	PD	2Lc-2Lab-2IL
Philosophy II *(Languages and Perceptions)*	H	Ph	2Lc-4IL

Table 10.2: Third Year: SB Stream (Spring Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Information Retrieval	CSE	IS, GV	2Lc-2Lab-2IL
Digital Signal Processing	CSE	CM, EE	2Lc-2Lab-2IL
Game Theory	CSE	IS, CN, AL	2Lc-2Lab-2IL
Network and Cyber Security	CSE	IAS, SP	2Lc-2Lab-2IL
Security-Blockchain project	CSE		
Elective I			
Elective II			
Elective III			

Table 10.3: Fourth Year: SB Stream (Fall Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Microcontrollers & Embedded Hardware	CSE	OS, PL	2Lc-2Lab-2IL
Distributed Ledger Technologies	CSE	IAS, SP	2Lc-2Lab-2IL
Foundations of Mathematical Economics and Quantitative Finance	CSE	SP	2Lc-2Lab-2IL
Elective IV			
Academic Research and Writing Culture	H	SP	

7. Economics
8. Distributed programming
9. Algorithms and Complexity
10. Analysis
11. Presentation
12. Communication
13. Technical Writing
14. Technical Reading
15. Problem Solving
16. Teamwork
17. Leadership
18. Professional Ethics
19. Privacy and Civil Liberties
20. Social context

10.3 Structure of the Two Years of Specialization

During the third and fourth year of this stream, the students take a set of core courses -.[1] Tables 10.1, 10.2, 10.3, and 10.4 show the list of courses for this stream. These tables also show the main areas and the knowledge areas covered by each course, along with its teaching structure.

Students also take electives course in these years, starting from the second semester

[1] Some of the courses are shared with other streams.

Table 10.4: Fourth Year: SB Stream (Spring Semester)

Course Name	Main Area	Knowledge Area	Teaching Structure
Business Analytics	CSE	SP	2Lc-2Lab-2IL
Elective V			
Elective VI			
Life Safety	H	SP	

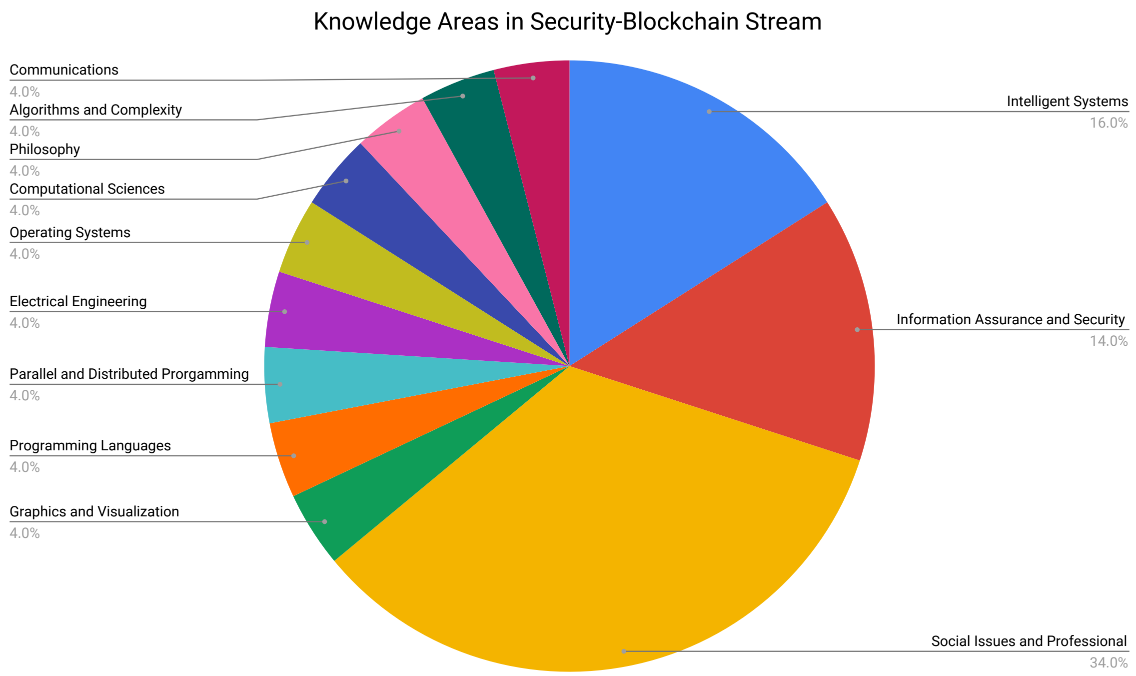

Figure 10.1: Breakdown of the knowledge areas (%) in the Security-Blockchain stream.

(spring semester) of the third year. While students can choose any electives they want from various available courses (the details of the electives are presented in Chapter 12), at least one of the three electives must be about a technical subject, and another about soft-skill, humanities, or economics/marketing/finance. In addition, students should do elective on a Security-Blockchain project supervised by Innopolis University faculty or Industry in the second semester of the third year. Finally, students must complete thesis work in the fourth year.

Summarized information about the Security-Blockchain stream is shown in Figure 10.1 and Table 10.5. Figure 10.1 shows the breakdown of the knowledge areas covered in the stream, and Table 10.5 shows the courses covering each knowledge area.

Table 10.5: Mapping between knowledge areas and courses in the Security-Blockchain stream

Knowledge	Courses	Portion
Intelligent Systems	*Information Theory* *Introduction to Machine Learning* *Information Retrieval* *Game Theory*	16 %
Information Assurance and Security	*Principles of computer security* *System and network administration* *Distributed Ledger Technologies* *System and network administration*	14 %
Social Issues & Professional Practice	*Fundamentals of computer security* *System and network administration* *Distributed Ledger Technologies* *Academic Research & Writing* *System and network administration* *Foundations of Mathematical Economics and Quantitative Finance* *Business Analytics* *Life Safety*	34 %
Graphics and Visualization	*Information Theory*	4 %
Programming Languages	*Microcontrollers & Embedded Hardwar*	4 %
Parallel and Distributed Programming	*Distributed Systems*	4 %
Electrical Engineering	*Digital Signal Processing*	4 %
Operating Systems	*Microcontrollers & Embedded Hardware*	4 %
Computational Sciences	*Game Theory*	4 %
Philosophy	*Philosophy* 2 (Languages and Perceptions)	4 %
Algorithms and Complexity	*Game Theory*	4 %
Communications	*Digital Signal Processing*	4 %

11 CORE COURSES

The following is the list of the core courses. For more details refer to Appendix L on page 53.

1. Introduction to Programming I
2. Computer Architecture
3. Discrete Mathematics
4. Mathematical Analysis I
5. Analytical Geometry & Linear Algebra – I
6. Philosophy I (Logic)
7. Introduction to Programming II
8. Data Structures and Algorithms
9. Mathematical Analysis II
10. Analytical Geometry & Linear Algebra – II
11. Theoretical Computer Science
12. Probability and Statistics
13. Physics I (mechanics)
14. Differential Equations
15. Data Modeling and Databases I
16. Operating Systems
17. Probability and Statistics
18. History
19. Data Modeling and Databases 2
20. Networks
21. Control Theory
22. Introduction to Artificial Intelligence
23. Software Project
24. Information Theory
25. Software Architectures
26. Introduction to Machine Learning
27. Distributed Systems and Cloud Computing
28. Compiler Construction
29. Philosophy II – Language & Perception
30. Introduction to Big Data
31. Theoretical Mechanics
32. Introduction to Robotics
33. Fundamentals of Computer Security
34. System and Network Administration
35. Information Retrieval
36. Digital Signal Processing
37. Data Mining
38. Game Theory
39. Project (SE, DS, AIR)
40. Software Systems Design
41. Lean Software Development
42. Nonlinear Control Theory
43. Robotic Systems
44. Sensors and Sensing
45. Mechanics and Machines
46. Network and Cyber Security
47. Statistical Techniques for Data Science
48. Numerical Modelling
49. Mathematical Modeling
50. Introduction to Computer Vision
51. Micro controllers and Embedded Hardware
52. Distributed Ledgers Technologies
53. Practical Machine Learning and Deep Learning

54. Software Quality and Reliability
55. Mobile Application Development
56. Mechatronics
57. Business Analytics
58. English for Academic Purposes I
59. English for Academic Purposes II

12 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

Part III

Structure of the MS Degree

SUMMARY OF THE MS DEGREE

The Master of Science degrees offered by the Faculty of Computer Science and Engineering at Innopolis University aim at creating professional software engineers, data scientists, robotics engineers, network engineers, and junior scientists who possess a deeper understanding of fundamental theoretical and practical results of computer science and engineering, are able to handle the complex abstractions to elaborate data, can use modern tools, languages and technologies, and understand the underlying human, social and industrial aspects of information technology.

Moreover, a person with one of such degrees acquired at Innopolis University should be eminently qualified to enter the IT job market in Russia in the position of software developer, software engineer, robotics engineer, data scientist, network engineer, or equivalent.

The primary target of these graduates is the industry of the city of Innopolis, therefore, the program foresees ample opportunities for synergies with such industry.

The Master of Science degrees provide a deeper treatment of specialized areas than the Bachelor of Science degree, equipping graduates to work with greater autonomy on more complex problems and to produce more innovative solutions.

The curricula last 2 years and are organized with courses distributed in 1 or 2 years and a substantial amount of research and internship activities. Courses are held mostly in English.

The following degrees are available:

- Software Engineering
- Data Science
- Robotics
- Secure Systems and Network Engineering

To provide a solid grounding in the application of the academic studies, in each year of study there is either an internship or a project in cooperation with a company or another institution, prioritizing those located inside the city of Innopolis.

Degrees are offered with a research, a practical, or a job orientation. At the end of the research-oriented and job-oriented degrees, the student writes a thesis. This thesis reflects the highest worldwide standards of university education and can also be developed in collaboration with a company.

13 Premises and organization

13.1 Premises

The Master of Science degrees offered by the Faculty of Computer Science and Engineering at Innopolis University aim at providing its students a quality graduate education in both the theoretical and applied foundations of computer science. The goal is to train students through comprehensive educational programs, and research in collaboration with industry and government, to effectively apply this education to solve real-world problems and enhance the graduates' potential for high quality, lifelong careers.

The degrees are organized with three different "orientations:"

- research oriented,
- practical oriented,
- job oriented.

The curricula include courses, projects, internships, and theses. Such curricula comprise a total of 120 ECTS (Europe Credit Transfer and Accumulation System) and the courses are performed mostly in English.

The **research orientation** focuses of providing the traditional instruction mechanisms, where first the students take classes, then gradually their blend their learning activities with research in the lab, and eventually perform a research centered thesis, expected to result in an advance of the scientific discipline.

The **practical orientation** is based on providing during the first semester an intensive course-based instruction, then in the second semester mixes courses instruction with a group project with real industrial customers and academic mentors, and finally have has the last two semesters based on internships in "real" companies providing a "real" and fully paid industrial experience.

The **job orientation** alternates coursework with job experience in "real" companies (a prerequisite for the admission), focuses on having students experimenting at work the material learnt in classes, and concludes with a empirical thesis in the effect of applying research concepts in real industrial environments.

The distribution of the coursework for the different curricula is presented in Table 14.2 on page 81.

There are seven Master degrees:

- Data Science
- Robotics
- Secure Systems and Network Engineering (SNE)
- Secure Systems and Network Engineering Research-Oriented (SNE-RO)
- Software Engineering (SE)
- Software Engineering Research-Oriented (SE-RO)
- Software Engineering Job-Oriented (SE-JO)

They are currently offered in the orientations described in Table 13.1.

Table 13.1: Current offering of MS degrees per orientation

Semester	Research Oriented	Practical Oriented	Job Oriented
Data Science	Currently running		
Robotics	Currently running		
SE	Starting this year	Currently running	Under consideration
SNE	Starting this year	Currently running	

13.2 Organization of the information about the MS Degree

The information about the MS Degree is organized as follows. Chapter 14 on page 74 describes the overall structure of the degree in terms of distribution of credits and courses. Then there are four chapters discussing the programs in the four areas: Chapter 15 on page 83 for Software Engineering, Chapter 16 on page 85 for Robotics, Chapter 17 on page 86 for Data Science, and Chapter 18 on page 88 for Secure Systems and Network Engineering. Finally, chapter 19 on page 90 presents in detail all the curricula and chapter 20 on page 103 describes in depth the core courses. In Section 14.7 on page 79, there is a table summarizing all the programs.

14 STRUCTURE OF THE DEGREES

14.1 Credits

Overall, the degrees require the student to acquire 120 credits over two years of academic work.

One credit of study approximately corresponds to 36 academic hours[1], of effort per student, which can be organized differently depending on the course, as detailed further in Section 14.4. This means that a course of 4 credits requires an effort of 144 hours, and an internship of 8 credits requires 288 hours (about 7 weeks).

Students will be required to attend core and elective courses (with technical content and with managerial/humanistic content), according to the regulations of the specific programs.

14.2 Organization of studies

14.2.1 Organization of studies for the practical-oriented, 1-year long curriculum

The organization of studies for the practical-oriented, 1-year long curriculum is based on 2 semesters with a substantial amount of internship and projects done in conjunction with partner companies.

The specifics of the studies depend on the kind of degree, while there are two major variations:

Variant 1 The courses of the curriculum span two semesters (Fall and Spring) and include core courses that provide the basic common knowledge to the students and elective courses that provide specialized knowledge according to the interest of the students. Core and elective courses are distributed throughout the year. While core courses are designed to provide a solid and common basic knowledge to all the students, elective course are specialized and defined every semester on the basis of the current

[1] In this document the terms “academic hour” and just “hour” are used interchangeably and are equivalent to 45 astronomical minutes.

technological trends and needs from industry. Moreover, the project work is developed in cooperation with industrial partners.

- 1st semester
  - Typically the student attends 4 core courses (see Section 20 on page 103)
  - The student attends some elective courses
- 2nd semester
  - Typically the student attends 4 core courses (see Section 20 on page 103)
  - The student attends some elective courses
  - The industrial project starts

Variant 2 The curriculum spans over 6 blocks (blocks are aligned to be 2 per semester, each block is 2 months except for the Research Project 1). Each block is 7 weeks long and two courses along with extensive lab sessions are covered in each block. After the first two blocks, a Research Project is carried out by the students in the 3rd block and is supervised by the IU faculty. The details of each block are as follows.

- Theory: 2 courses (4+4 hours)
- Practice: 2 courses (8+8 hours)

At the end of the 5th block, the students undertake a research project with industry where real-world research problems are posed by industry beforehand. The students work in groups and are supervised by both a representative from industry and an IU faculty member. SNE students take 8 core courses (see Section 20 on page 103) and 2 elective courses during the program.
Formally, the successful students will receive the degree two academic years after starting the program.

14.2.2 Organization of studies for the research-oriented, 2-years long curriculum

The organization of studies for the research-oriented, 2-years long curriculum is based on 4 semesters of studies. Each semester is split into 2 blocks.

During each year a student is required to take 30 credits of studies per semester. After 1st year in summer for 4 weeks the student is required to perform internships either in research labs or in companies that are partners of Innopolis at the discretion of the supervisor. During the 2nd semester of second year the student dedicates 2 days a week for internship.

The progress of studies is as follows:

- Admission and induction at the start of the academic year:
  - Upon admission, a student is assigned to a lab of Innopolis University and to a supervisor drawn from among the full time professors of Innopolis University affiliated with that lab. The assignment is made by the Department of Education, after
    * consultation with the Dean and the Associate Dean for Teaching,
    * consultation with he heads of labs and the supervisors who have expressed an interest in accepting the student in question.
  - Typically, the student is immediately introduced to research activities. The student may apply for a change of supervisor up to the end of the first semester and the decision is taken by the Department of Education, after consulting with the current supervisor, the lab head, the institute head, the associate dean for teaching, and the dean.
  - Before the beginning of the first academic year there is a mandatory bootcamp, explaining the principle of research methods, of organization of work, and related issues such as IP, plagiarism, among others.
- First year:
  - Typically a student takes 4 courses in the first semester, and 4 courses in the second semester.
  - The supervisor may identify "knowledge deficits" and consequently may assign extra courses, which, in this case, can also be BS courses.
  - At the end of the second semester, the student finalizes the topic for the final thesis.
  - It is also required that during the time between semesters the student is actively involved in an internship in a research lab or in a corporate partner of Innopolis University, at the discretion of the supervisor.
- Second year:
  - During the second year the student is required to take elective courses and work on their thesis, as advised by their supervisor.
  - Also during the second year it is expected that during the time between semesters the student is actively involved in an internship either in a research lab or in a corporate partner of Innopolis University, at the discretion of the supervisor.
  - Additional credits can be awarded via individual studies undertaken by the student with the supervisor.

  - In case of specific needs of the research work, the supervisor may identify further "knowledge deficits" and consequently may assign extra courses, which, in this case, can also be BS courses, individual studies, or courses taken in summer schools or other events.

The defence of the thesis typically occurs during the month of August of the second year. The evaluation of the thesis is done according to the policies set forth in the university regulations.

14.2.3 Organization of studies for the job-oriented, 2-years long curriculum

The organization of studies for the job-oriented, 2 year curriculum is based on 4 semesters of studies performed in conjunction with job activities in a partner company.

The specifics of the studies depend on the kind of degree.

14.3 Main Areas and Knowledge Areas

To effectively design the curricula, we identify areas that must be covered. We distinguish the following two kinds of areas, i.e., main areas and knowledge areas. First, the main areas are the key components of the degree program listed as follows:

- (CSE) Computer Science and Engineering
- (M) Math
- (P) Physics
- (H) Humanities
- (M-CS) Math and Computer Science
- (IT) Internship and Theses

Meanwhile, the knowledge areas shown in the following list identify more specifically the body of knowledge that must be covered in the curricula.

- (IS) Intelligent Systems
- (GV) Graphics and Visualization
- (PL) Programming Languages
- (PD) Parallel and Distributed Programming
- (SDF) Software Development Fundamentals

- (SE) Software Engineering
- (EE) Electrical Engineering
- (CE) Control Engineering
- (OS) Operating Systems
- (CN) Computational Science
- (MP) Mathematical Physics
- (ME) Mechanical Engineering
- (AL) Algorithms and Complexity
- (HCI) Human Computer Interaction
- (SP) Social Issues and Professional Practice
- (CM) Communications

We identified the knowledge areas based on the Curriculum Guidelines for Undergraduate Degree Programs in Computer Science by the joint task force between ACM (Association for Computing Machinery) and IEEE Computer Society in 2013.[2] Note that a knowledge area is a higher level concept than a course, and an individual course can cover several knowledge areas when appropriate.

We provide throughout the document how each course is mapped to a main area and knowledge areas. We also provide the distribution of the knowledge areas in the different programs.

14.4 Organization of the courses of instruction

Courses can be organized in a variety of ways, taking advantage of various teaching mechanisms, such as:

L: Classroom Lecture;

T: Classroom Tutorial, where the instructor explains concepts guiding students through practical exercises;

La: Group Laboratory, where the students perform homeworks and/or exercises and/or projects coordinated by the instructor, asking help from the instructor when needed, and receiving additional material directly from the instructor

IL: Individual labs, where the students perform exercises and refer to the instructor in person or via electronic media to get support in case of need.

[2] Association for Computing Machinery (ACM) Joint Task Force on Computing Curricula and IEEE Computer Society. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. ACM, New York, NY, USA, 2013. ISBN 978-1-4503-2309-3. 999133

In terms of scheduling, typically the core courses for the masters program can be organized in two ways:

- One semester long, with 4 academic hours of instruction per week
- Half semester (7 weeks) long, with 8 academic hours of instruction per week, and 4 additional hours spread through the semester at the discretion of the supervisor

Where possible, the courses are offered during the evening hours to accommodate working students.

Additional courses can be offered in the following format, especially in cases where the instructor is a person from outside Innopolis:

- 3 or 4 weekends throughout the semester
- one or two full “blocked” week(s)

14.5 Number of admitted students

The number of admitted students is defined by the Department of Education based on the requests of the senior university administration and of the university stakeholders.

For the research-oriented 2 year programs, every full time professor will typically supervise from 4 to 12 master’s students across the two years of study; exceptions can be made by the Department of Education upon request of the interested professor, after hearing the institute head, the associate Dean for teaching, and the dean.

14.6 Summary of the competences

See Table 14.1.

14.7 Synoptic view of the curricula

See Table 14.2.

Table 14.1: Competences and curricula

Competence	Curriculum						
	DS	SNE	SNE-RO	R	SE	SE-RO	SE-JO
	2Y	1Y	2Y	2Y	1Y	2Y	2Y
Industrial experience		X			X	X	X
Understanding of the research challenges	X		X	X		X	
Use, develop, and design complex systems					X	X	X
Software architecture					X	X	X
Requirements collection and analysis					X	X	X
Different development models					X	X	X
Software testing					X	X	X
Empirical and statistical models	X				X	X	X
Statistics, machine learning, and deep learning	X				X	X	X
Design and develop a cyber-physical system				X			
Programming industrial and mobile robots				X			
Developing control algorithms for robotic systems				X			
Working with sensors and embedded systems				X			
Working with vision systems				X			
Developing human-robot interactions				X			
Use, develop, and design centralized and distributed databases	X						
Technologies and tools for big data	X				X	X	X
Data mining	X						
Data visualization	X						
Randomized algorithms for data science	X						
Data mining and information retrieval	X						
Computer and network security		X	X				
Network architectures and protocols		X	X				
Design and development of distributed and scalable systems		X	X				
Computer forensics and cyber crime investigations		X	X				
Assessing vulnerabilities in IT systems		X	X				
Cryptography		X	X				
Penetration Testing		X	X				

Table 14.2: Overview for all programs

Semester	SE 1Y-PO	SE 2Y-RO	DS 2Y-RO	R 2Y-RO	SNE 1Y-PO	SNE 2Y-RO
Fall 1	Metrics and empirical methods			Sensing, perception, and actuation	Inter-Networking and Routing	
	Requirements Engineering		Big data technologies and analytics	Dynamics of non-linear robotic systems	Classical Internet Applications	
	Managing Software Development		OOP and Architecture	Elective (Tech)	Security of Systems and Networks	
	Communication		Machine Learning		Advanced Networking	
	Personal Software Process		Elective (Hum)	Elective (Tech)	Research Project	
	Elective (Hum)		DS Project	Thesis	Elective (Hum)	
Spring 1	Models of Software Systems		Advanced Statistics	Computer Vision	Large Systems	
	Analysis of Software Artifacts		Advanced Machine Learning	Behavioral and Cognitive Robotics	Cyber-crime and Forensics	
	Architectures of Software Systems		Advanced Information Retrieval	Advanced Robotics	Offensive Technologies	
	Industrial Project		Elective (Tech)	Elective (Tech)	Advanced Security	
			Elective (Hum)	Project		
			Thesis	Thesis		
			Internship			
Summer 1	Elective (Tech)	Thesis		Internship	Industrial Project	Thesis
	Elective (Tech)			Thesis		
	Industrial Project					

Semester	SE		DS	R	SNE	
	1Y-PO	2Y-RO	2Y-RO	2Y-RO	1Y-PO	2Y-RO
Fall 2	Internship	Machine Learning	Optimization	Computational Intelligence	Internship	Elective (Hum)
		Elective (Tech)	High-Dimensional Data Analysis	Elective (Tech)		Elective (Tech)
		Thesis	Elective (Tech)	Elective (Hum)		Thesis
			Thesis	Thesis		
Spring 2	Internship	Elective (Tech)	Elective (Tech)	Neuroscience	Internship	Elective (Hum)
		Elective (Tech)	Elective (Hum)	Elective (Tech)	Thesis	Elective (Tech)
		Thesis	Thesis	Elective (Hum)		Thesis
				Thesis		
Summer 2	Thesis			Thesis		

15 MS IN SOFTWARE ENGINEERING

15.1 Educational goals of the programs

Software systems are becoming increasingly complex and ubiquitous. The current trend in the development of IT services and devices is not only related to an increased complexity due to the functionalities required by the customers but also due to the shift of many features from a hardware implementation to a software one. This last aspect has huge benefits in terms of costs and flexibility but has drawbacks in terms of complexity. The main benefits are related to the standardization of the hardware (implying lower production costs), the ability to deliver over-the-air-updates (implying the possibility of adding new features and fixing issues continuously), and shifting the main differentiation among products from hardware to software. However, the increased amount of software in such a wide variety of devices and services requires a new kind of professional, able to manage such complexity in terms of products and development processes. Such professionals are required to design and develop highly complex systems in environments where the requirements are volatile and the development teams are of various sizes, skills, and temperaments, and often geographically distributed.

The practical-oriented one-year curriculum focuses on providing hard and soft skills that can be applied immediately in industrial projects. Particular attention is given to the applicability of the theoretical concepts and their usage in practice. Moreover, the curriculum includes an industrial project developed in a team with an industrial customer, in which the students have the opportunity to apply the concepts and skills they have developed under the continuous supervision of both an academic and an industrial mentor. At the end of the program, students can be employed immediately and be productive in an industrial environment.

The research-oriented two-year curriculum focuses on providing the same hard and soft skills of the practical-oriented one-year curriculum with additional advanced topics provided in the second year and a research-based thesis. The first year is completely shared with the practical-oriented one-year curriculum while in the second year students are able to select electives and work on their thesis.

The job-oriented two-year curriculum focuses on providing the same hard and soft skills of the practical-oriented one-year curriculum but with a different pace, allowing students to have a job position and experiment the skills acquired in the course in their real life projects.

15.2 Key competences and practices for the students

All MSIT-SE graduates will have the following competencies after successful completion of the course of studies.

1. Advanced skills to use, develop, and design complex software systems
2. Mastery in software architecture
3. Proficiency in requirements collection and analysis
4. Full understanding of different development models
5. Expertise in software testing
6. Understanding of complex empirical and statistical models
7. Comprehensive understanding of the technical and non-technical issues related to the development of complex systems
8. Ability to monitor the software process
9. Skills in designing and developing a complex system
10. Practical experience and understanding of an industrial environment

16 MS IN ROBOTICS

16.1 Educational goals of the program

Robotics is considered as a multidisciplinary General Purpose Technology and is the best tool for Project Based Learning (PBL) for the 21st century and the most exciting way to learn by doing. While working on robotics projects, students acquire deeper knowledge through active exploration of real-world challenges and problems and expand their knowledge in the related areas. The goal of the program is to impart key knowledge in the field of robotics and major related disciplines and to create professionals who are able to solve complex multidisciplinary problems, constantly expanding their knowledge and skills.

The research-oriented program aims at preparing highly motivated students to continue their career by pursuing a PhD, to work in a robotics R&D position at a company, or to start a new business based on robotics-related new products or solutions.

16.2 Key competences and practices for the students

Master in Robotics students will acquire the following competences

1. Working in a team
2. Performing challenging tasks in robotics and AI
3. Programming industrial and mobile robots
4. Developing control algorithms for robotic systems
5. Working with sensors and embedded systems
6. Working with vision systems
7. Modeling and prediction of robot behaviors
8. Developing frameworks and algorithms for Human-computer/robot interaction
9. Developing new robotics solutions with elements of AI
10. Carrying out research in a specific area of robotics
11. Presenting results and ideas for research and industrial community

17 MS IN DATA SCIENCE

17.1 Educational goals of the programs

The explosion of the Internet has increased by orders of magnitude the amount of data available to managers, decision and policy makers, investors, researchers, and also ordinary people. It has become therefore a necessity to elaborate techniques to manage such amounts of data and to transform it into useful, and possibly actionable, information. To create professionals able to handle such important and complex issues, Innopolis University has established the MS in Data Science, whose primary goal is to educate future scientists, managers, and entrepreneurs on one side on how to store and retrieve vast amounts of data locally and distributedly, in a secure, efficient, and effective way, and, on the other, on how to extract and interpret the information present in such data, and use it to make reliable assessments and predictions taking advantage of the latest technologies coming from advanced statistics, data mining, and machine learning. As a side goal of this program, students are educated in the fundamental aspects of programming and of mathematics underlying the competence above. Furthermore, throughout the program special attention is given in applications in areas related to finance, banking, transportation, security, software engineering, etc.

The research-oriented two-year curriculum places particular emphasis on providing the students with sophisticated competences in statistics, machine learning, data mining, databases, and information retrieval, that enable the student to continue her or his studies toward a doctoral degree or to gain employment in R&D labs of industrial companies. Moreover, the program includes a thesis aimed at fully mastering a novel aspect of data science, providing innovative insights and new results in some unexplored area.

17.2 Key competences and practices for the students

The MS-DS graduates will have the following competencies after successful completion of the course of studies.

1. Skills to use, develop, and design centralized and distributed databases and technologies and tools for big data

2. Proficiency in data mining
3. Full understanding of statistics, machine learning and deep learning
4. Comprehensive knowledge of data mining and information retrieval
5. Advanced mastery in statistics, experimental design, machine learning, and deep learning
6. Sophisticated understanding of algorithms for data science
7. Understanding of research problems

18 MS IN SECURE SYSTEMS AND NETWORK ENGINEERING

18.1 Educational goals of the program

The practical-oriented 1 year SNE curriculum emphasizes the concrete real world problems faced by industry, mastering concrete hard and soft skills needed to secure information both through cryptographic and non-cryptographic techniques. Furthermore, the program also focuses on network security, cyber security, web application security, applied cryptography, secure development, configuration, and so forth. A research project is carried out in collaboration with industry where the students work on a real world research problem provided by industry. Such practice on the one hand showcase our students to industry, and on the other hand expose the students to the industrial environment. Additionally SNE graduates acquire hands-on experience of data security, communication security, wireless network security, Bluetooth security, information forensics, large networks administration, offensive technologies, and so on.

The research-oriented 2 year SNE curriculum provides the same skills of the practical-oriented one-year curriculum but with a set of additional courses and a research thesis. The first year completely overlaps with the practical-oriented curriculum allowing the students to select the specific program during the Spring semester of the first year.

18.2 Key competences and practices for the students

The MS-SNE graduates will have the following competencies after successful completion of the course of studies.

1. Knowledge of computer security, including security protocols and cryptography
2. Understanding of the architecture and protocols of classical Internet applications (DHCP, DNS, the Email system, WWW)
3. Understanding of many of the intricacies related to designing and developing a distributed computer systems

4. Understanding of the challenges of building a scalable IT infrastructure that is flexible and efficient to manage (Virtualization and Cloud Computing, Workstation and Server deployment, Datacenters and Infrastructure management methodologies)
5. Insight and skills in the subject of practical IT security (Physical security, Network security, Database security, Application security, Web-apps security)
6. Expertise in conducting computer forensics and cyber crime investigations
7. Knowledge in identifying and assessing vulnerabilities in IT systems including computers, networks and applications software
8. Expertise in designing and building security systems and their components
9. Applied cryptography and implementation of cryptographic primitives for system, network, and cyber security
10. Network Traffic and Security Analysis
11. Network Security Management
12. Penetration Testing
13. Webapp security analysis
14. Research skills

19 DETAILED CURRICULA OF STUDY

19.1 Software Engineering (practice-oriented, one-year)

The practice-oriented software engineering program is organized in three semesters: Fall, Spring, and Summer. During the two semesters 7 core courses will be provided:

- Metrics and empirical methods for software engineers and data scientists
- Requirements Engineering
- Managing Software Development
- Models of Software Systems
- Personal Software Process
- Analysis of Software Artifacts
- Architectures for Software Systems
- Communication

Students can take elective courses throughout the year.

Students also need to complete internship activities and an industrial project supervised by IU faculty and an industrial mentor throughout the year.

Summarized information is presented in Table 19.1.

Furthermore, knowledge areas covered by each course are presented in Table 19.2 while Table 19.3 shows the mapping between knowledge areas and courses. Finally, Figure 19.1 shows the overall breakdown.

Table 19.1: Software engineering 1-year.

Fall	Spring	Summer
Metrics and empirical methods for software engineers and data scientists (5 ECTS)	Models of Software Systems (5 ECTS)	Elective (Tech) (5 ECTS)
Requirements Engineering (5 ECTS)	Analysis of Software Artifacts (5 ECTS)	Elective (Tech) (5 ECTS)
Managing Software Development (5 ECTS)	Architectures for Software Systems (5 ECTS)	Industrial Project (9 ECTS)
Communication (2 ECTS)	Industrial Project (5 ECTS)	
Personal Software Process (2 ECTS)		
Elective (Hum) (2 ECTS)		

Table 19.2: Knowledge areas of software engineering 1-year.

Course	Main Area	Knowledge Area	Teaching Structure
Requirements Engineering	CSE	SE	2Lc-2Lab-2IL
Models of Software Systems	CSE	SE	2Lc-2Lab-2IL
Managing Software Development	CSE	SE	2Lc-2Lab-2IL
Metrics and empirical methods for software engineers and data scientists	CSE	SE, AL	2Lc-2Lab-2IL
Personal Software Process	CSE	SE	2Lc-2Lab-2IL
Analysis of Software Artifacts	CSE	SDF, SE, AL	2Lc-2Lab-2IL
Architectures for Software Systems	CSE	SE, HCI	2Lc-2Lab-2IL
Communication	CSE	SP, Ph	2Lc-2Lab-2IL

Table 19.3: Mapping between knowledge areas and courses in the software engineering 1-year.

Knowledge Area	Course	Portion
Software Development Fundamentals	Analysis of Software Artifacts Metrics and empirical methods for software engineers and data scientists	5%
Software Engineering	Requirements Engineering Personal Software Process Models of Software Systems Managing Software Development Metrics and empirical methods for software engineers and data scientists Analysis of Software Artifacts Architectures for Software Systems	78%
Algorithms and Complexity	Models of Software Systems Analysis of Software Artifacts	4%
Human Computer Interaction	Architectures for Software Systems	1%
Social Issues and Professional Practice	Communications	10%
Philosophy	Communications	3%

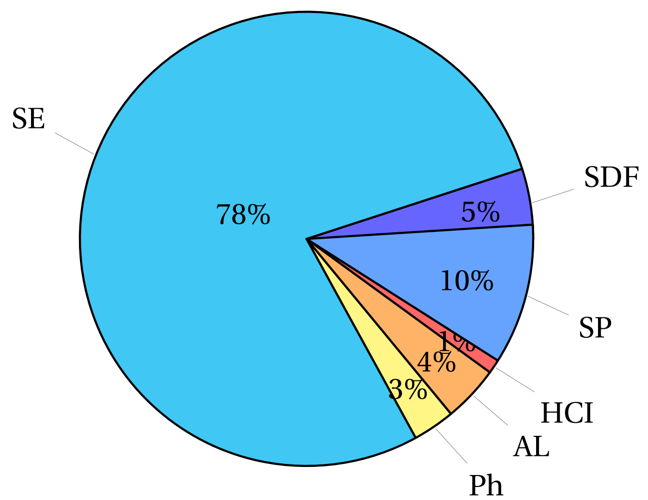

Figure 19.1: Knowledge areas of software engineering 1-year.

Table 19.4: Software engineering research-oriented 2-year: year 1

Spring
Industrial project (9 ECTS)

Table 19.5: Software engineering research-oriented 2-year: year 2

Fall	Spring
Machine Learning (Tech) (5 ECTS)	Elective (Tech) (5 ECTS)
Elective (Tech) (5 ECTS)	Elective (Tech) (5 ECTS)

19.2 Software Engineering (research-oriented, two-year)

The research-oriented two-year software engineering program is organized in five semesters (three in the first year and Fall and Spring in the second year). The first year completely overlaps with the practice-oriented one-year program for the first and second semesters (Table 19.1) with a different organization starting from the Summer of the first year (Table 19.4). The second year provides the opportunity to the students to attend elective courses (Table 19.5) and work on a research-based thesis.

Knowledge areas of the program and relative mapping between knowledge areas and courses is exactly as in the practice-oriented one-year program (Section 19.1).

19.3 Software Engineering (job-oriented, two-year)

The job-oriented two-year software engineering program is organized in four semesters. During the four semesters the same number of core courses (8) of the practice-oriented one-year program will be provided. Courses will be offered in the timeframe 17:30-20:30 to allow students working in companies to attend, with a particular focus on Innopolis companies. The second year of full-internship is now allocated over two years 8:30-17:00 like a normal working day. For the students to have time to prepare assignments and exams the delivery of courses is concentrated over three days, typically Monday-Wednesday. In Table 19.6 the first year is presented. Fall semester has three courses to be thought every evening for 4 academic hours (3 solar hours). Thursdays and Fridays are left for homework. The spring semester has two courses Monday and Tuesdays and the rest of the week is left for self-study and mostly for the academic project. The second year works similarly as shown in Table 19.7 with the industrial project instead of the academic project in the Spring semester. More space is left to electives. With respect to the classic program here class allocation is preferably contiguous with one specific course thought of any day. Knowledge areas of the program and relative mapping between knowledge areas and courses is exactly as in the practice-oriented

Table 19.6: Software engineering job-oriented 2-year: year 1

Fall	Spring	Summer
Metrics and empirical methods for software engineers and data scientists (5 ECTS)	Analysis of Software Artifacts (5 ECTS)	Elective (Tech) (5 ECTS)
Managing Software Development (5 ECTS)	Architectures for Software Systems (5 ECTS)	Elective (Tech) (5 ECTS)
Communication (2 ECTS)		
Personal Software Process (2 ECTS)		

Table 19.7: Software engineering job-oriented 2-year: year 2

Fall	Spring
Requirements Engineering (5 ECTS)	Models of Software Systems (5 ECTS)
Elective (Hum) (2 ECTS)	

one-year program, only split over two years.

19.4 Software Engineering: Project across courses

This is a proposal for the Academic Project that appears in Fall Semester for the one-year program and in Spring of the first year for the proposed two-year program.

The SE program is characterized by an industrial project with a strong impact and a strong relevance and by academic projects typically defined and developed locally to each course. For a project to be pedagogically effective, and for students to learn concepts integrated into the development process, it is necessary to have a project running across all the courses of the Fall semester of the first year (or Spring for the job-oriented version).

The different core courses cover different aspects and phases of the software development process, so it comes natural to think about an integrated project where each course cover (and evaluate) a specific aspect. Here we define a general structure that should be instantiated every year differently to avoid students just presenting artifacts of previous years' colleagues:

- **PHASE 1**: Requirements are collected in natural language in *Managing Software Development* where the lecturer and TAs could play the role of stakeholders.
- **PHASE 2**: Requirements are specified using diagrammatic and/or formal notations in *Models of Software Systems* with the support of lecturer and TAs
- **PHASE 3**: The system can now be designed, implemented, tested and implemented in *Requirements engineering* using a diagrammatic notation like UML and a specific technological stack determined by the simulated business domain and by the lectures and TAs of the course.

- **PHASE 4**: Software quality can then be measured and assessed in *Metrics and empirical methods for software engineers and data scientists* and *Analysis of software artifacts*.

To a large extent this project is defined according to a "waterfall model", but can also implemented in a more iterative way repeating the different phases through sprints. The important part is the synchronization between courses:

- **PHASE 1**: The requirements have to be determined by **week 4**, and expressed in a natural-language-based notation. The related course has to approach this topics early in the course.
- **PHASE 2**: The formal specification has to be produced by **week 8**, the necessary notion of the course have to be presented by then
- **PHASE 3**: The system has to be designed and implemented by **week 12**, the related course has to arrange topics accordingly
- **PHASE 4**: The artifacts has to be evaluated and measured before the final exam at **week 15**

Clearly the output artifact of each phase may need some adjustment while proceeding to the next phase. DMD project has a similar approach within the same course: requirements, ER diagrams, relational schema, query implementation and interface implementation. It is waterfall, and it is necessary to operate some adjustment to all the artifacts before proceeding to the next.

Table 19.8: Robotics 2-years long. Year 1

Fall	Spring
Sensing, perception, and actuation (5 ECTS)	Computer Vision (5 ECTS)
Dynamics of Nonlinear Robotic Systems (5 ECTS)	Behavioral and cognitive robotics (5 ECTS)
Machine Learning (5 ECTS)	Advanced Robotics (5 ECTS)
Elective (Tech) (5 ECTS)	Elective (Tech) (5 ECTS)
Elective (Tech) (5 ECTS)	Project (5 ECTS)

Table 19.9: Robotics 2-years long. Year 2

Fall	Spring
Computational intelligence (5 ECTS)	Neuroscience (5 ECTS)
Elective (Tech) (5 ECTS)	Elective (Tech) (5 ECTS)
Elective (Hum) (2 ECTS)	Elective (Hum) (2 ECTS)

19.5 Robotics (research-oriented, two-year)

The research-oriented robotics program is organized in four semesters, Fall and Spring over two years. The following 8 core courses will be provided:

- Sensation, perception, and actuation
- Advanced Robotics
- Dynamics of Nonlinear Robotic Systems
- Machine Learning
- Computer Vision
- Computational Intelligence
- Behavioral and cognitive robotics
- Neuroscience

Students also need to complete internship activities throughout the two years and a thesis starting from the Fall semester of the first year.

Summarized information is presented in Table 19.8 and 19.9.

Furthermore, knowledge areas covered by each course are presented in Table 19.10 while Table 19.11 shows the mapping between knowledge areas and courses. Finally, Figure 19.2 shows the overall breakdown.

Table 19.10: Knowledge areas of robotics 2-years.

Course	Main Area	Knowledge Area	Teaching Structure
Sensation, perception, and actuation	CSE	IS, EE, CE	2Lc-2Lab-2IL
Advanced Robotics	CSE	IS, CE, ME	2Lc-2Lab-2IL
Dynamics of Nonlinear Robotic Systems	CSE	IS, CE, ME	2Lc-2Lab-2IL
Computer Vision	CSE	IS, GV, AL	2Lc-2Lab-2IL
Machine Learning	CSE	IS, AL	2Lc-2Lab-2IL
Computational intelligence	CSE	IS, CE, AL	2Lc-2Lab-2IL
Behavioral and cognitive robotics	CSE	IS, AL	2Lc-2Lab-2IL
Neuroscience	CSE	IS, CE, AL	2Lc-2Lab-2IL

Table 19.11: Mapping between knowledge areas and courses in the robotics 2-years.

Knowledge Area	Course	Portion
Intelligent Systems	Sensation, perception, and actuation Advanced Robotics Dynamics of Nonlinear Robotic Systems Computer Vision Machine Learning Computational Intelligence Behavioral and cognitive robotics Neuroscience	50%
Graphics & Visualization	Computer Vision	3%
Electrical Engineering	Sensation, perception, and actuation	5%
Control Engineering	Sensation, perception, and actuation Advanced Robotics Dynamics of Nonlinear Robotic Systems Computational Intelligence Neuroscience	14%
Mechanical Engineering	Advanced Robotics Dynamics of Nonlinear Robotic Systems	10%
Algorithms and Complexity	Machine Learning Computer Vision Computational Intelligence Behavioral and cognitive robotics Neuroscience	19%

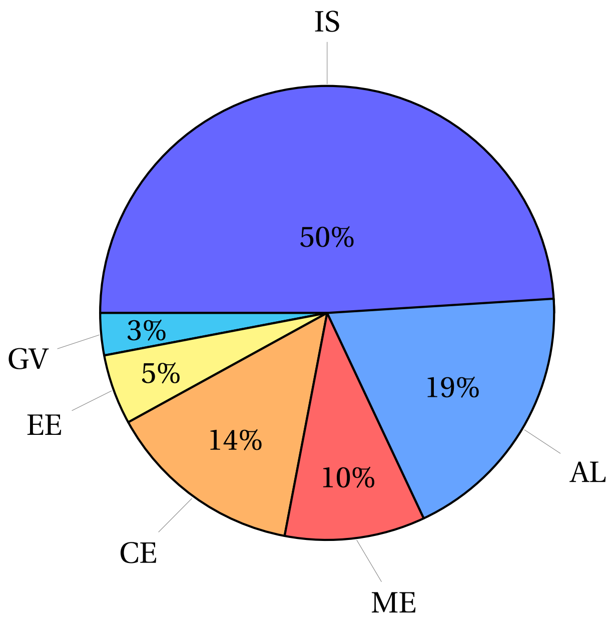

Figure 19.2: Knowledge areas of robotics 2-years.

19.6 Data Science (research-oriented, two-year)

The research-oriented data science program is organized into four semesters, Fall and Spring over two years. The following 9 core courses will be provided:

- Metrics and empirical methods for software engineers and data scientists
- High-Dimensional Data Analysis
- Machine Learning
- Optimization
- Advanced Information Retrieval
- Big Data Technologies and Analytics
- Advanced Statistics
- Advanced Machine Learning
- Managing Software Development

Students also need to complete internship activities throughout the two years and a thesis starting from the Fall semester of the first year.

Summarized information is presented in Tables 19.12 and 19.13.

Furthermore, knowledge areas covered by each course are presented in Table 19.14 while Table 19.15 shows the mapping between knowledge areas and courses. Finally, Figure 19.3 shows the overall breakdown.

Table 19.12: Data science 2-years. Year 1

Fall	Spring
Metrics and empirical methods for software engineers and data scientists (5 ECTS)	Advanced Statistics (5 ECTS)
Big Data Technologies and Analytics (5 ECTS)	Advanced Machine Learning (5 ECTS)
OOP and Software Architecture (5 ECTS)	Advanced Information Retrieval (5 ECTS)
Machine Learning (5 ECTS)	Elective (Tech) (5 ECTS)
Elective (Hum) (2 ECTS)	Elective (Hum) (2 ECTS)
DS Project (5 ECTS)	Thesis

Table 19.13: Data science 2-years. Year 2

Fall	Spring
High-Dimensional Data Analysis (5 ECTS)	Elective (Tech) (5 ECTS)
Optimization (5 ECTS)	Elective (Hum) (2 ECTS)
Elective (Tech) (5 ECTS)	
Thesis	Thesis

Table 19.14: Knowledge areas of data science 2-years.

Course	Main Area	Knowledge Area	Teaching Structure
Metrics and empirical methods for software engineers and data scientists	CSE	SE, AL	2Lc-2Lab-2IL
High-Dimensional Data Analysis	CSE	IS, GV, AL	2Lc-2Lab-2IL
Advanced Information Retrieval	CSE	AL	2Lc-2Lab-2IL
Big Data Technologies and Analytics	CSE	IS, AL	2Lc-2Lab-2IL
Advanced Statistics	CSE	CN	2Lc-2Lab-2IL
Machine Learning	CSE	CN, AL	2Lc-2Lab-2IL
Advanced Machine Learning	CSE	CN, AL	2Lc-2Lab-2IL
Optimization	CSE	IS, CN, AL	2Lc-2Lab-2IL
OOP and Software Architecture	CSE	SE	2Lc-2Lab-2IL

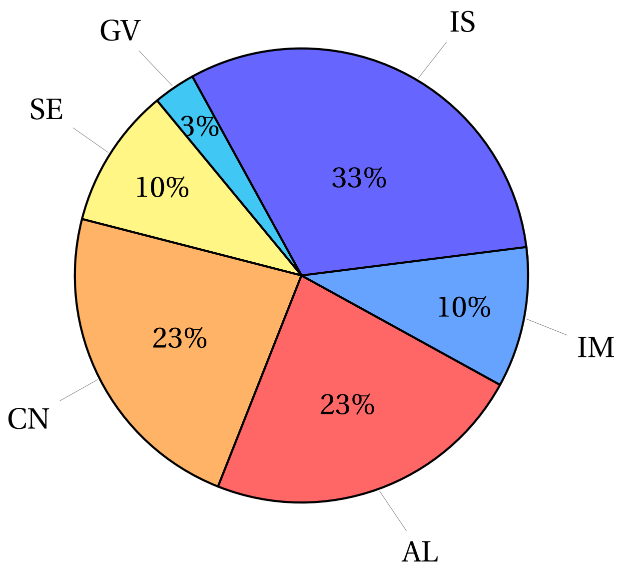

Figure 19.3: Knowledge areas of data science 2-years.

Table 19.15: Mapping between knowledge areas and courses in the data science 2-years.

Knowledge Area	Course	Portion
Intelligent Systems	High-Dimensional Data Analysis Machine Learning Optimization Advanced Information Retrieval Big Data Technologies and Analytics Advanced Machine Learning	33%
Graphics & Visualization	High-Dimensional Data Analysis	3%
Software Engineering	Metrics and empirical methods for software engineers and data scientists OOP and Software Architecture	10%
Computational Science	Advanced Statistics Machine Learning Optimization Advanced Information Retrieval Big Data Technologies and Analytics Advanced Machine Learning	23%
Algorithms and Complexity	Metrics and empirical methods for software engineers and data scientists High-Dimensional Data Analysis Machine Learning Optimization Advanced Information Retrieval Big Data Technologies and Analytics Advanced Machine Learning	23%
Information Management	Advanced Information Retrieval Big Data Technologies and Analytics	10%

19.7 Secure Systems and Network Engineering (practice-oriented, one-year)

The MS-SNE graduates will have the following competencies after successful completion of the course of studies.

- Knowledge of computer security, including security protocols and cryptography,
- Understanding of the architecture and protocols of classical Internet applications (DHCP, DNS, the Email system, WWW),
- Understanding of many of the intricacies related to designing and developing a distributed computer systems,
- Understanding of the challenges of building a scalable IT infrastructure that is flexible and efficient to manage (Virtualization and Cloud Computing, Workstation and Server deployment, Datacenters and Infrastructure management methodologies),
- Insight and skills related to practical IT security (Physical security, Network security, Database security, Application security, Web-apps security),
- Expertise in conducting computer forensics and cyber crime investigations
- Knowledge in identifying and assessing vulnerabilities in IT systems including computers, networks and applications software, and
- Expertise in designing and building security systems and their components.

The practice-oriented, 1 year SNE curriculum emphasizes the concrete real world problems faced by industry, mastering concrete hard and soft skills needed to secure information both through cryptographic and non-cryptographic techniques. The program also focuses on network security, cyber security, web applications security, applied cryptography, secure development, configuration, and so forth. A research project is carried out in collaboration with industry during which the students work on real world research problems provided by industry. Such practice on the one hand showcases our student to industry, on the other hand exposes the students to the industrial environment. Additionally, SNE graduates acquire hands-on experience of data security, communication security, wireless network security, Bluetooth security, information forensics, large networks administration, offensive technologies, and so on. Details of the courses are given in Table 19.16

SNE graduates of the practice-oriented, 1 year curriculum will acquire the following competences in addition to the above competencies:

- Applied cryptography and implementation of cryptographic primitives for system, network, and cyber security

Table 19.16: Curriculum for 1 year SNE program.

	Aug-Sep-Oct	Oct-Nov-Dec	Dec-Jan
1st Semester	**Block 1**	**Block 2**	**Block 3**
	Classical Internet Applications (6 ECTS)	Security of Sys. & Nets. (6 ECTS)	Project (4 ECTS)
	Inter-Networking and Routing (INR) (6 ECTS)	Advanced Networking (6 ECTS)	Elective (Hum) (2 ECTS)
	Jan-Feb-Mar	**Mar-Apr-May**	**May-Jun-Jul-Aug**
2nd Semester	**Block 4**	**Block 5**	**Block 6**
	Advanced Security (6 ECTS)	Offensive Technologies (6 ECTS)	Project (5 ECTS)
	Large Systems (5 ECTS)	Cybercrimes and Forensics (6 ECTS)	Elective (Hum) (2 ECTS)

Table 19.17: Knowledge areas of SNE 1-year.

Course	Main Area	Knowledge Area	Teaching Structure
Inter-Networking and Routing	CSE	AL, CM	2Lc-2Lab-2IL
Classical Internet Applications	CSE	AL, CM	2Lc-2Lab-2IL
Advanced Networking	CSE	PD, OS, AL, CM	2Lc-2Lab-2IL
Security of System and Networks	CSE	AL, CM	2Lc-2Lab-2IL
Cybercrimes and Forensics	CSE	CN, CM	2Lc-2Lab-2IL
Advanced Security	CSE	OS, AL, CM	2Lc-2Lab-2IL
Large Systems	CSE	PL, OS, CN, AL, CM	2Lc-2Lab-2IL
Offensive Technologies	CSE	PL, OS, AL, CM	2Lc-2Lab-2IL

- Network Traffic and Security Analysis
- Network Security Management
- Penetration Testing
- WebApp security analysis
- Research skills

Furthermore, knowledge areas covered by each course are presented in Table 19.17 while Table 19.18 shows the mapping between knowledge areas and courses. Finally, Figure 19.4 shows the overall breakdown.

19.8 Secure System and Network Engineering (SNE-RO) (research-oriented, two-years)

The research-oriented two-year Secure System and Network Engineering program is organized in four semesters. The first year completely overlaps with the practice-oriented one-year program (Table 19.16). The second year provides the opportunity to the students to attend elective courses (Table 19.19) and work on a research-based thesis.

Knowledge areas of the program and relative mapping between knowledge areas and courses is exactly as in the practice-oriented one-year program (Section 19.7).

Table 19.18: Mapping between knowledge areas and courses in the SNE 1-year.

Knowledge Area	Course	Portion
Programming Languages	Large Systems Offensive Technologies	3%
Parallel & Distributed Programming	Large Systems	1%
Operating Systems	Inter-Networking and Routing Advanced Security Large Systems Offensive Technologies	8%
Computational Science	Cybercrimes and Forensics Large Systems	4%
Algorithms and Complexity	Advanced Networking Classical Internet Applications Distributed Systems Security of System and Networks Advanced Security Large Installation Administration Offensive Technologies	30%
Communications	Advanced Networking Classical Internet Applications Distributed Systems Security of System and Networks Cybercrimes and Forensics Advanced Security Large Installation Administration Offensive Technologies	54%

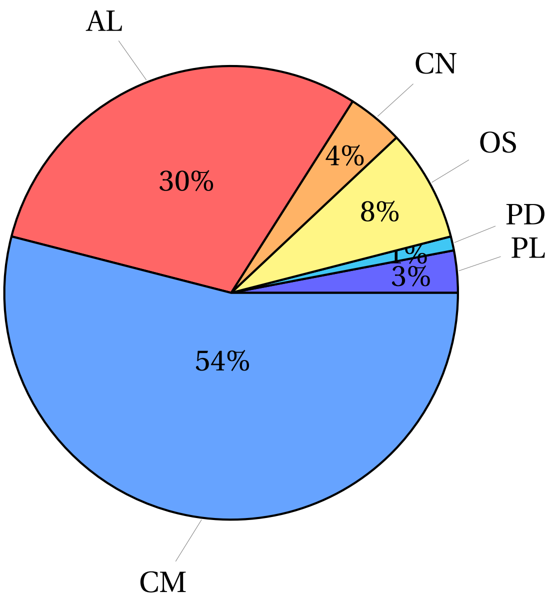

Figure 19.4: Knowledge areas of SNE 1-year.

Table 19.19: Secure System and Network Engineering research-oriented 2-year: year 2

Fall	Spring
Elective (Tech) (5 ECTS)	Elective (Tech) (5 ECTS)
Elective (Tech) (5 ECTS)	Elective (Tech) (5 ECTS)

20 CATALOGUE OF THE CORE COURSES

The core courses administered in the programs are:

1. Advanced Information Retrieval (see Section **??** on page **??**)
2. Advanced Machine Learning (see Section **??** on page **??**)
3. Advanced Networking (see Section **??** on page **??**)
4. Advanced Robotics (see Section **??** on page **??**)
5. Advanced Security (see Section **??** on page **??**)
6. Advanced Statistics (see Section **??** on page **??**)
7. Analysis of Software Artifacts (see Section **??** on page **??**)
8. Architectures for Software Systems (see Section **??** on page **??**)
9. Behavioral and Cognitive Robotics (see Section **??** on page **??**)
10. Big Data Technologies and Analytics (see Section **??** on page **??**)
11. Classical Internet Applications (see Section **??** on page **??**)
12. Communication (see Section **??** on page **??**)
13. Computational Intelligence (see Section **??** on page **??**)
14. Computer Vision (see Section **??** on page **??**)
15. Cybercrime and Forensics (see Section **??** on page **??**)
16. Distributed Systems (see Section **??** on page **??**)
17. Dynamics of Nonlinear Robotic Systems (see Section **??** on page **??**)
18. High-Dimensional Data Analysis (see Section **??** on page **??**)
19. Large Installation Administration (see Section **??** on page **??**)
20. Machine Learning (see Section **??** on page **??**)
21. Managing Software Development (see Section **??** on page **??**)
22. Metrics and Empirical Methods for Software Engineers and Data Scientists (see Section **??** on page **??**)
23. Models of Software Systems (see Section **??** on page **??**)
24. Neuroscience (see Section **??** on page **??**)
25. Offensive Technologies (see Section **??** on page **??**)
26. Optimization (see Section **??** on page **??**)
27. Personal Software Process (see Section **??** on page **??**)
28. Requirements Engineering (see Section **??** on page **??**)

29. Security of System and Networks (see Section ?? on page ??)
30. Sensing, Perception, and Actuation (see Section ?? on page ??)

They are divided across programs as follows:

- **Software Engineering**
  1. Analysis of Software Artifacts (see Section ?? on page ??)
  2. Architectures for Software Systems (see Section ?? on page ??)
  3. Communication (see Section ?? on page ??)
  4. Managing Software Development (see Section ?? on page ??)
  5. Requirements Engineering (see Section ?? on page ??)
  6. Metrics and Empirical Methods for Software Engineers and Data Scientists (see Section ?? on page ??)
  7. Models of Software Systems (see Section ?? on page ??)
  8. Personal Software Process (see Section ?? on page ??)
- **Robotics**
  1. Advanced Robotics (see Section ?? on page ??)
  2. Behavioral and Cognitive Robotics (see Section ?? on page ??)
  3. Computational Intelligence (see Section ?? on page ??)
  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 ?? on page ??)
  8. Sensing, Perception, and Actuation (see Section ?? on page ??)
- **Data Science**
  1. Advanced Information Retrieval (see Section ?? on page ??)
  2. Advanced Machine Learning (see Section ?? on page ??)
  3. Advanced Statistics for DS (see Section ?? on page ??)
  4. Big Data Technologies and Analytics (see Section ?? on page ??)
  5. High-Dimensional Data Analysis (see Section ?? on page ??)
  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 ?? on page ??)
- **Secure Systems and Network Engineering**

1. Advanced Networking (see Section **??** on page **??**)
2. Classical Internet Applications (see Section **??** on page **??**)
3. Distributed Systems (see Section **??** on page **??**)
4. Security of System and Networks (see Section **??** on page **??**)
5. Cybercrime and Forensics (see Section **??** on page **??**)
6. Advanced Security (see Section **??** on page **??**)
7. Large Installation Administration (see Section **??** on page **??**)
8. Offensive Technologies (see Section **??** on page **??**)

21 CATALOGUE OF THE ELECTIVE COURSES

The elective courses administered in the programs are:

1. Information Retrieval for Data Science (see Section **??** on page **??**)
2. ...

They are divided across programs as follows:

- **Software Engineering**
  1. ...
- **Robotics**
  1. ...
- **Data Science**
  1. Information Retrieval for Data Science (see Section **??** on page **??**)
  2. ...
- **Secure Systems and Network Engineering**
  1. ....

Part IV

Structure of the PhD Degree

22 SUMMARY OF THE PHD DEGREE

The PhD program in Computer Science (CS) implemented by the Innopolis University according to the legislation, rules and regulations of the Russian Federation, in general it spans 4 years and includes 4 principal components:

- Taught courses, which are in turn divided into:
  - Base (Core) courses which are common and mandatory for all PhD students irrespective of their specialization, which include both CS and non-CS courses;
  - Mandatory Specific CS courses
  - Optional Specific CS courses
- Practical component, which contains:
  - Instructional (Teaching) practice;
  - Professional (Skills and Work Experience) practice
- Scientific research leading to a PhD. This also includes writing up the PhD thesis, and preparing and publishing research papers on the topics connected to the PhD research.
- State Examination (mandatory)

23 TAUGHT COURSES: CURRICULA OF STUDY

23.1 Core Courses

- History and Philosophy of Science
- Foreign Language
- Informatics and Computer Facilities

23.2 Mandatory Specific Courses

- Theoretical Foundations of Computer Science
- Pedagogics and Psychology in Higher Education
- Contemporary Problems and Research Methods in Computer Science
- Research Methods and Methodology

23.3 Optional (Elective) Courses

- **Part 1:**
  - Advanced Design and Architectures for Software Systems
  - Open Issues in Statistical methods in Data Science
  - Design of Robotic Manipulation Systems
  - Advanced Cybercrimes and Computer Forensics
- **Part 2:**
  - Advanced Functional Programming in Haskell
  - Optimization of Security Systems
  - Design Methods in Behavioural and Cognitive Robotics
  - Generative methods in Machine Learning
  - Promoting quality of life for people with disabilities in the IT industry

24 PROFESSIONAL PRACTICE

24.1 Time Allocation

According to the current rules and regulations issued by the Russian Ministry of Science and Higher Education, the Professional Practice component must include:

- 68 hours of supervised practice, 496 hours of stand-alone (unsupervised) work and studies, and 12 hours of reporting and control procedures in Semester 3;
- 68 hours of supervised practice, 568 hours of stand-alone work and studies, and 12 hours of reporting and control procedures in Semester 5;
- 68 hours of supervised practice, 208 hours of stand-alone work and studies, and 12 hours of reporting and control procedures in Semester 8.

24.2 General Framework

In a view of a very significant amount of time allocated to the Professional Practice component, it is of paramount importance to implement this component in a way that will deliver the maximum value to the PhD program in general, to the development of professional skills and work experience of each PhD student, and to fulfill the Russian regulatory requirements at the same time.

Therefore, the Professional Practice component of the PhD programme should be closely correlated with the Taught Courses and with the PhD Research components. At the same time, Professional Practice should broaden up the content of the Taught Courses and to provide PhD students with hands-on skills and experiences which are deemed to be important for their future professional life.

For example, a hypothetical situation when a student successfully completes their PhD program in CS but is nevertheless unable to engineer and implement a commonplace software solution using any of the contemporary programming technologies, is considered to be unacceptable. One of the objectives of the Professional Practice is to make such unfortunate situations impossible.

The Professional Practice component should therefore be structured as a series of individual, personalised Computer Science / Software Engineering Projects (thereafter SE Projects).

The subjects and titles of these projects should correspond to the subjects of the Taught Courses, and to be sufficiently close (whenever possible) to the subject of the PhD research undertaken by a student.

The subject and the precise requirements of each Project are to be elaborated jointly by the PhD Supervisor and the Professor in charge of the corresponding Taught Course, taking into consideration the views and preferences of the PhD student, and to be approved by the PhD Department of the University.

The SE Projects can be undertaken within Innopolis University and/or external organizations, and should contain the elements of stand-alone work / self-studies by the PhD student, with the amount of time allocated to such work / studies as stipulated by the current Government regulations.

24.3 List of Taught Courses for which SE Projects may be Implemented

- Informatics and Computer Facilities (Core, taught in Semesters 1 and 2). Should be linked to an SE Project in Semester 3. Typically, the subject of this project would be to achieve proficiency in some programming language or software engineering technology (taking into account PhD student's preferences). For example, getting in-depth, hands-on experience with modern object-oriented programming and the software development and project management toolchains.
- Theoretical Foundations of Computer Science (Mandatory CS course, taught in Semester 4). May be linked to an SE Project in Semester 5. For example, the project may include foundations of Functional Programming which is a practical manifestation of modern principles of Theoretical Computer Science.
- Contemporary Problems and Research Methods in Computer Science (Mandatory CS course, taught in Semester 6). May be linked to an SE Project in Semester 5. Although such a project predates the taught course, the former can be used to identify and highlight the problems which are subsequently addressed in the taught course.
- The following Optional CS courses taught in Semester 5 should normally be linked with the corresponding SE Projects in Semester 5, thus providing a close integration between the theoretical and practical aspects of the subjects studied:
- Advanced Design and Architectures for Software Systems Modern Statistical Methods in Data Science Design of Robotic Manipulation Systems Advanced Cybercrimes and

Computer Forensics

- The Professors in charge of the above courses would be required to develop the corresponding programs for the SE Projects.
- Similarly, the following Optional CS courses taught in Semester 7 should normally be linked with the corresponding SE Projects in Semester 8:
  - Advanced Functional Programming in Haskell
  - Optimization of Security Systems
  - Design Methods in Behavioural and Cognitive Robotics
  - Generative methods in Machine Learning
  - Promoting quality of life for people with disabilities in the IT industry

24.4 Implementation Schedule

The Professional Practice program for the current PhD students at Innopolis University is to be implemented starting from Sprint Term of 2020. Those 2nd-year PhD students in the academic year 2019/2020 who did not complete their Professional Practice in Semester 3 (that is, in the Fall Term of 2019), must do so in the Spring Term of 2020,

25 FINAL STATE ATTESTATION

This document describes the general content of state examination, the thesis structure, the format and process of review and the conduct of the final defence. This is an ideal process subject to verification of legal requirements.

25.1 State examinations

1. History and Philosophy of Science (Oral Exam)
2. Foreign language (English) A paper in the English language should be submitted
3. Informatics and Computer Facility (Perform the “Second Year Exam”).

25.2 Format of the thesis

The actual content of the thesis is topic-specific and it has to be agreed between student and supervisor. Here we suggest a general structure that has to be customized according to the specific needs.

Title, abstract, acknowledgements, table of contents As an appendix, thesis cover and template is attached.

Chapter 1: Introduction

The context of investigation is presented and the specific open problem that is addressed is introduced. A typical narrative for this approach can be “outside-in”, i.e. from a general concept and context to the specificity of the open problem and its solution. The chapter 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 and the purpose of the thesis*
- *list the contributions*
- *layout the structure of the thesis*

Chapter 2: Literature Review

The state of the art for the topic investigated is surveyed. To successfully complete this chapter books, research journals, conference proceedings, handbooks, and online material should be consulted.. 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 the same approach before?*
- *Where is that work reported?*
- *What is the outline of your approach*

Chapter 3: Methodology

This chapter includes the general structure of the research: methods, approaches and processes followed. The methodology should be linked back to the aims of the thesis and the literature to explain why certain methods have been used. Experiments carried out should be explained. This chapter should answer:

- *What is your proposed (original) methodology?*
- *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?*

Chapter 4: Results, Analysis and Discussion

This chapter contains the overall results that have been achieved. Results have to be presented in a manner that makes explicit their relationship to the research question. Analysis and discussion of the findings should be included. Accuracy and relevance of the results have to be discussed and compared with other researchers' results. Are the results satisfactory? How success is defined?

Chapter 5: Conclusion

The consequences of the achieved results have to be discussed in this chapter. Are results satisfactory and how they can be improved? This chapter is a short account of the results of the work, emphasising what is new in it. The chapter should align with the Introduction, in which the problem was described. Limitations of the work should be addressed and suggest what further work might be done. Make a synthesis of the contributions and impact of the work and recommendations.

25.3 Assessment

The thesis should be assessed as pass/fail. [need to verify according to the laws of RF]. No conditional pass is allowed.

25.4 Anti-plagiarism

Anti-plagiarism tools can be used to identify overlapping with online resources. However, the tool-produced results have to be analysed by the commission and supervisors in order to determine their nature, sources and impact. Assessment of plagiarism cannot be machine-determined since a semantic analysis is necessary, and not syntactic. Plagiarism tools often identify overlaps with technical reports or draft appearing on Researchgate or Arxiv.org.

25.5 Supervisor and supervising committee

For each student a supervisor is appointed following a negotiation and mutual agreement between the two parties. Eligible supervisors are staff members satisfying the following three conditions:

PhD degree has to be awarded in Russia or, if obtained from a foreing country, should be approved in Russia via a process of nostrification Position should be "professor/docent" [very unclear, legal requirements should be verified].

In order to support an effective learning process, a committee including two more members apart from the supervisors is proposed by the supervisor and appointed by the Department of Education after consultation with the student and the supervisor. It is advisable that one of the two members belongs to the same research area (or a close one) of the supervisor and the student, while the second member could belong to a different area in order to inject new ideas in the creative process. While the principal supervisor has to satisfy the two conditions mentioned above, for the two members of the committee these may be relaxed. However, it is advisable when possible to appoint committee members that also satisfy the conditions. Under specific circumstances, that have to be approved by the faculty body, one (and only one) of the committee members could be a visiting professor, a representative of industry or someone external to the university.

25.6 Form for the review of the thesis

Reviews are conducted in free form by three reviewers who prepare a letter and send it to the thesis' supervisor and commission. Reviewers must not belong to the university and are proposed by the supervisor in consultation with the Department of Education. The letter should address the following points: Is the problem significant for the scientific community? What is the novelty of the solution or of the approach? Is the technical solution accurate? Is the problem solved leading to applications that can improve state-of-the-art technology or can have a societal or economic impact?

25.7 Defence

Once the thesis has been submitted and evaluated successfully by the reviewers a final defence can be arranged. The Department of Education appoints for the occasion:

1. A chair belonging to the university faculty
2. Three defense committee members, disjoint from the supervising committee, but with potential overlap with the reviewers in case the logistics could be arranged. The defense committee members must not be part of the supervising committee, nor can be the supervisor nor other members.

The final defense is in the English language and it is structured as follows:

- The candidate presents his work without interruption for 45 minutes in a public defense. [need to verify according to the laws of RF]
- Each of the committee members asks questions in a rotation under the coordination of exam chairs. Several rounds of questions can be conducted until no more question appear or on the chair's discretion
- In the end, other faculty members present could ask questions. Faculty members that are not part of the committee can attend and ask questions, but they do not participate in deliberations.
- The candidate and all non-committee members will leave the room and the commission will deliberate on success or failure of the defence. In case of disagreement, the chair will moderate the discussion and try to reach consensus. If consensus is not reached, a vote will be cast between the three defense committee members. The chair must not vote. The deliberation will be according to majority (two out of three).

BIBLIOGRAPHY

Association for Computing Machinery (ACM) Joint Task Force on Computing Curricula and IEEE Computer Society. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. ACM, New York, NY, USA, 2013. ISBN 978-1-4503-2309-3. 999133.