Course Descriptions - NJIT & UO

Mathematics Part

5 Courses

Advanced Mathematics B I (NJIT Transferred)

Discrete Structures (UO)

Advanced Mathematics B II (NJIT)

Linear Algebra A (NJIT)

Probability and Maths-phsics Sta.A (NJIT)

The explanation for the suffixes

My degree programme is a joint degree programme between the University of Oulu (UO) and Nanjing Institute of Technology (NJIT). Therefore, the courses are either delivered by UO or NJIT.

1. "NJIT Transferred" stands for the course that was delivered by NJIT and its credits have already been converted into ECTS credits by UO.
2. "NJIT" stands for the course that was delivered by NJIT and because UO has not converted them yet, the credits need to be converted into ECTS by multiplying 1.5 according to the guidelines provided on the programme website.

   ECTS – University of Copenhagen (ku.dk)
3. "UO" stands for the course that was delivered by UO and its credits are already ECTS credits.

Name

Advanced Mathematics B I

ECTS Credits (NJIT Transferred)

5 ECTS credits / 133 hours of work

Language of instruction

Chinese

Timing

1st year, 1st semester

Learning outcomes

After completing the course, the student will be able to:

- understand the concepts of function and master its notation. Know the parity, monotonicity, periodicity, and boundedness of function. Understand the concepts of composite functions, inverse functions, and implicit functions. Master the properties and graphics of basic elementary functions. Understand the concept of limit and the limits of segmented functions. Master the four arithmetic operations and the method of using the two important limits to find the limits. Master the two criteria for the existence of limits and use them to find the limits. Understand the concepts of infinitesimal, infinite, and infinitesimal order, and use equivalent infinitesimals to find the limit. Understand the concept of continuity and different types of discontinuity points in the function. Know the continuity of elementary functions, the properties of continuous functions on closed intervals, and how to apply these properties.
- understand the concept of derivatives, the geometric and physical meaning of derivatives, and the relationship between the derivability and continuity of functions. Master the four arithmetic operations of derivatives, the derivative of the compound functions, and the derivative formula of basic elementary functions. Be able to find the first and second-order derivatives of segmented functions. Understand the concept of higher-order derivatives and find the nth-order derivatives of simple functions. Master how to find the second derivative of elementary functions. Be able to find the first and second-order derivatives of functions determined by implicit functions and parametric equations. Know the concept of differentiation, the four arithmetic operations, and use derivatives to describe some simple physical quantities.
- understand and be able to apply Rolle’s theorem and Lagrange's theorem. Know Cauchy's theorem. Understand the concept of the extreme value of a function, master the method of using derivatives to discriminate the monotonicity of a function, and methods to find the extreme value of a function. Master the method of using L’ Hospital law to find the limit of indeterminate form.
- understand the concept and properties of indefinite integral. Master the basic formulas of indefinite integrals, the method of integration by substitution, and the method of integration by parts.
- understand the basic concept of definite integral, the mean value theorem for definite integrals. Understand the variable limit function and its derivative theorem. Master the Newton-Leibniz formula. Master the properties of definite integrals, the method of integration by substitution, and the method of integration by parts. Know the concept and how to calculate anomalous integrals.
- master the differential element method for definite integrals and the applications of definite integrals on geometry, such as calculating the area of plane figures, the arc length of a plane curve, the volume of the solid of revolution, and the volume of the solid with the known parallel section.

Contents

The main contents of the Advanced Mathematics B course contain the basic concepts, theories, arithmetic operations, and related applications. It gradually develops the abilities for students to generalize problems, logical reasoning, spatial imagination, and more proficient computing. In this way, students are trained in the methods of mathematical analysis and the use of these methods to solve real-world problems, laying the necessary mathematical foundation for the study of subsequent courses.

- Chapter 1: Function and Limit
    1. Mapping and Functions
    2. Limit of Sequences
    3. Limit of Functions
    4. Operation Laws of Limit Calculation, Infinitesimal and Infinite
    5. Criteria for Judging the Existence of Limit and Two Important Limits
    6. Infinitesimal Comparison
    7. Continuity and Discontinuity of Functions, Operations of the Continuous Functions, Continuity of Elementary Functions, and Properties of Continuous Functions on A Closed Interval
- Chapter 2: Derivative and Differential
    1. Concept of Derivative
    2. Derivative Methods
    3. Differentiation of Functions
    4. Higher-order Derivative
    5. Implicit Function and Its Derivative Determined by the Parametric Equation
- Chapter 3: Differential Mean Value Theorem and Applications of Derivative
    1. Differential Mean Value Theorem
    2. L’ Hospital Law
    3. Taylor’s Formula
    4. Monotonicity of Functions, and Concavity and Convexity of Curves
    5. Extremum, Maximum, and Minimum of Functions
    6. Graphical Depiction of Functions
- Chapter 4: Indefinite Integral
    1. Concept and Properties of Indefinite Integrals
    2. Integration by Substitution
    3. Integration by Parts
    4. Integral of Rational Function and Uses of Integral Table
- Chapter 5: Definite Integral
    1. Concept and Properties of Definite Integrals
    2. Basic Formula of Calculus
    3. Definite Integral’s Integration by Substitution and by Parts
    4. Anomalous Integrals
- Chapter 6: Applications of Definite Integral
    1. Differential Element Method for Definite Integrals
    2. Applications of Definite Integrals on Geometry

Mode of delivery

Contact teaching

Learning activities and teaching methods

Contact teaching, homework, exam.

Recommended or required reading

Weng Liangui, Wu Qingkuan: Advanced Mathematics Part2. ISBN: 978-7-044221-2.

Assessment methods and criteria

exercises, course attendance, exam.

Name

Discrete Structures

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

1st year, 2nd semester

Learning outcomes

After completing the course, the student will be able to:

- Interpret simple mathematical notation and write it by himself/herself
- Define the most important concepts that appear in the basic theoretical structures (see Contents) accurately and describe the ideas of the definitions
- Master the most essential results and (algorithmic) methods of the basic structures and is capable of applying them in easy examples
- Evaluate certain kinds of simple practical problems in computer science, choose appropriate theoretical methods to solve the problems, and finally apply the chosen methods

Contents

1. The concept of algorithm 2. Number systems and the conversion of bases 3. Logic (proposition calculus, predicate calculus basics) 4. Set theory, relations, and functions (finiteness/infinity, induction) 5. Basic number theory (divisibility and the respective algorithms) 6. Combinatorics (principles of counting) 7. Graph theory

Mode of delivery

Contact teaching

Learning activities and teaching methods

Lectures, exercises, group work, independent work

Prerequisites and co-requisites

Scientific Communication

Recommended or required reading

Lecture slides (about 250 slides), lecture notes (about 100 pages), textbook: Peter Grossman, Discrete Mathematics for Computing, Second Revised Edition, Palgrave Macmillan, 2002. ISBN: 978-0333981115.

Assessment methods and criteria

Active participation and accepted assignments

Grading

Numerical scale 1-5 or fail

Person responsible

Pertti Karhapää

Name

Advanced Mathematics B II

NJIT Credits (NJIT)

4 NJIT credits / 6 ECTS credits (converted) / 160 hours of work

Language of instruction

Chinese

Timing

1st year, 2nd semester

Learning outcomes

After completing the course, the student will be able to:

- understand the concept of the vector and its notation. Master vector operations (linear operation, scalar product, vector product), know the condition that two vectors are perpendicular and parallel. Know the mixed product of vectors. Know the coordinate expression of the unit vector, vector, direction number, and direction cosine. Be able to use the coordinate expression to do vector operations. Master plane equations (point-normal form, intercept form, general form), linear equations (parametric equation, symmetric equation, and general equation), and be able to solve relevant problems with the relationship between plane lines. Understand the concept of the surface equation. Know the equations and graphics of common quadric surfaces. Be able to solve the problems related to the surface of revolution with coordinate axis as the rotation axis and the cylindrical equation of which with generatrix being parallel to coordinate axis. Understand the parametric equation and general equation of space curve. Understand its projection on the coordinate plane and be able to solve its equation.
- understand the concept of multivariate functions; Understand the concepts of limit and continuity of functions of two variables and the properties of continuous functions on bounded closed regions; Understand the concepts of partial derivatives and total derivatives, understand the necessary and sufficient conditions for the existence of total derivatives, and know that how to apply total derivatives in approximate calculation; master the methods to calculate the first and second-order partial derivative of function composition and to find the partial derivatives of implicit functions (including implicit functions determined by system of equations); Understand the tangent and normal planes of curves, understand the concept of the tangent and normal planes of surfaces, and know how to find their equations; understand the concept of the extreme value and the conditional extreme value of the multivariate function, grasp the necessary conditions for the existence of the extreme value of the multivariate function, and know the sufficient conditions of the existence of the extreme value of the functions of two variables. Be able to calculate the extreme value of a function of two variables, use the Lagrange multiplier method to calculate the conditional extreme value, find the maximum and minimum values of simple multivariate functions, and solve some simple application problems.
- understand the concept of the double integral and its properties. Master the calculating methods of double integrals (in the Cartesian coordinate system and polar coordinate system). Use multiple integrals to calculate some geometric and physical quantities (such as area of plane figures, volume, and mass of solid figures, etc.).
- understand the concept and properties of the two types of curve integrals and the relationship between them. Master the methods to calculate the two types of curve integrals. Master Green's formula, be able to use the condition that the plane curve integral is independent of the path and be able to find the primitive

function of total derivatives; Be able to use curve integral to calculate some geometric and physical quantities (arc length and mass).

- understand the concepts of differential equations and solutions, general solutions, initial conditions, and particular solutions; Master the equation with separable variables and how to calculate the solution of the first-order linear equation; Be able to solve homogeneous equations and solve some differential equations with simple variable substitution; Be able to use the method of reducing order to solve the following equation: $y(n) = f(x), y'' = f\ (x, y'), y'' = f(y, y')$; Understand the properties of solutions of linear differential equations and the structure theorem of general solutions; Master the solution of second or higher order homogeneous linear differential equations with constant coefficients; Be able to find the special solutions of second-order non-homogeneous linear differential equations with constant coefficients (the free term is composed of polynomials, exponential functions, sinusoidal functions, cosine functions, and their sum and product); be able to solve some simple application problems with differential equations.

Contents

- Chapter 8: Space Analytic Geometry and Vector Algebra
  1. Vector and Its Linear Operation
  2. Scalar Product and Vector Product
  3. Curved Surface and Its Equation
  4. Space Curves and Its Equation
  5. Plane and Its Equation
  6. Straight Lines in Space and Its Equation
- Chapter 9: Differentiation of Functions Several Variables and Its Application
  1. Basic Concept of Functions of Several Variables
  2. Partial Derivative
  3. Total Derivative
  4. Derivation Rule of Functions of Several Variables
  5. Derivation Formula for Implicit Functions
  6. Geometric Application of Differentiation of Function of Several Variables
  7. Extreme Value of Function of Several Variables and Its Solution
- Chapter 10: Multiple Integrals and Its Application
  1. Concept of Double Integrals and Its Properties
  2. Evaluation of Double Integrals
  3. Applications of Multiple Integrals
- Chapter 11: Line Integrals and Surface Integrals
  1. Line Integrals of the First Type
  2. Line Integrals of the Second Type
  3. Green’s Formula and Its Application
- Chapter 12: Differential Equation
  1. Concept of Differential Equation
  2. Separable Differential Equation
  3. Homogeneous Equation
  4. Linear Differential Equation of the First Order
  5. High-order Differential Equations Turned to Lower-order Differential Equation
  6. Linear Differential Equation of Higher-order
  7. Homogeneous Linear Differential Equation with Constant Coefficient

8. Non-homogeneous Linear Differential Equation with Constant Coefficient

Mode of delivery

Contact teaching

Learning activities and teaching methods

Contact teaching, homework, exam.

Recommended or required reading

Weng Liangui, Wu Qingkuan: Advanced Mathematics Part2. ISBN: 978-7-044221-2.

Assessment methods and criteria

exercises, course attendance, exam.

Name

Linear Algebra A

NJIT Credits (NJIT)

2 NJIT credits / 3 ECTS credits (converted) / 80 hours of work

Language of instruction

Chinese

Timing

2nd year, 1st semester

Learning outcomes

After completing the course, the student will be able to:

- understand the basic knowledge of determinants, matrix, linear equation, eigenvalue, and eigenvector. Master the fundamental theories and methods of linear algebra. Be able to apply these methods for mathematical modeling and problem-solving.
- transform the practical problem into mathematical models, especially linear models and their solutions. Lay the necessary mathematical foundation for related follow-up courses. Understand the status and development trends related to the application of linear algebra, so that will be equipped with the abilities of continuous learning and foresight.

Contents

Linear Algebra is an important basic theoretical course for science, engineering, economics, and management, etc. It plays a critical role, not only in the specialties of mathematics, mechanics, and physics, but also in the majors of computer graphics, computer-aided design, and cryptography. The main contents of this course are determinant, matrix and its operations, the elementary transformation of the matrix, the linear correlation of vector system and linear system of equations, eigenvalues and eigenvectors of matrices, etc. Through the course, students will acquire abilities of higher abstract thinking, logical reasoning, and more proficient arithmetic ability. Students will master the orderly, algebraic, linear, and solvable processing methods to the research objects, so that lay the necessary mathematical foundation for further study, including the fields of computer and other majors like scientific research and innovative new technologies.

- Chapter 1: Determinant
    1. Second-order and Third-order Determinants
    2. Definition of N-th Order Determinant
    3. Properties of Determinant
    4. Cramer’s Rule and Its Applications
- Chapter 2: Matrix
    1. Concept of Matrix and Its Operations
    2. Inverse Matrix
    3. Block Matrix
    4. Elementary Transformation of Matrix
    5. Rank of Matrix
- Chapter 3: Linear Relation of Vector Groups and Solution of Linear Equation Groups
    1. Judgment of Solutions to Linear Equations
    2. Linear Relation of Vector Groups (Linearly Dependent and Linearly Independent)
    3. Rank of Vector Group

4. Structure of Solutions to Linear Equation Groups

- Chapter 4: Eigenvalue, Similarity Matrix, and Quadratic Form
  1. Inner Product of Vectors, and Orthogonality of Vectors
  2. Eigenvalue and Eigenvector of Matrix

Mode of delivery

Contact teaching

Learning activities and teaching methods

Contact teaching, homework, exam.

Recommended or required reading

Weng Liangui, Wu Qingkuan: Advanced Mathematics Part2. ISBN: 978-7-044221-2.

Assessment methods and criteria

exercises, course attendance, exam.

Name

Probability and Maths-physics Sta.A

NJIT Credits (NJIT)

3 NJIT credits / 4.5 ECTS credits (converted) / 120 hours of work

Language of instruction

Chinese

Timing

2nd year, 2nd semester

Learning outcomes

After completing the course, the student will be able to:

- establish the idea of randomness, recognize the universality of the existence of random phenomena, the versatility of application, and the importance of the course. Be able to create new methods for understanding uncertain systems in the reality. Can develop the new model of thinking under the era of big data.
- get basic knowledge of the formation process of the probability axiomatization through the learning of the probability part. Master the basic knowledge and solutions to common problems such as typical random variables and their distributions and corresponding numerical characteristics.
- master the basic theories and methods of statistics, create statistic models to address the specific problem through the learning of mathematical statistics. Be able to apply these abilities that have been obtained from the course to deal with the complex problems related to engineering.

Contents

Probability Theory and Mathematical Statistics is a compulsory basic mathematics course for students in the majors of science and engineering. It provides a set of fundamental mathematic approaches for quantitative analysis of random phenomena and random data. The course contains the probability theory part and mathematical statistics part. The probability Theory part is based on the modern mathematical framework to the statistical regularity of random phenomena. The Mathematical Statistics part studies how to collect and utilize the data that have been influenced randomly, and then, make the reasonable estimate and inference on the target question that is under research so that further to give more constructive reference and suggestions.

- Chapter 1: Basic Concept of the Probability Theory
  1. Random Experiment
  2. Frequency and Probability
  3. Classic Models of Probability
  4. Conditional Probability and Independence
- Chapter 2: Random Variable and Probability Distributions
  1. Random Variable and Probability Distributions
  2. Distribution Functions of Random Variables
  3. Continuous Random Variables and Their Probability Density
  4. Distribution of Functions of Random Variables
- Chapter 3: Multidimensional Random Variable and Distribution
  1. Two Dimensional Random Variables
  2. Marginal Distribution and Conditional Distribution
  3. Independent Random Variables

    4. Distributions of Functions of Random Variables

- Chapter 4: Numerical Characteristics of Random Variables
    1. Mathematical Expectation and Variance
    2. Covariance and Correlation Coefficient
    3. Covariance Matrix
- Chapter 5: Numerical Characteristics of Random Variables
    1. Law of Large Numbers
    2. Central Limit Theorem
- Chapter 6: Sampling and Sampling Distributions
    1. Population and Sample and Statistic
    2. Sampling Distributions
- Chapter 7: Parameter Estimation
    1. Point Estimate and Evaluating Estimators
    2. Interval Estimate
    3. Interval Estimate of Population Mean and Variance in Normal Distribution
    4. Interval Estimate of Population Parameters in Non-normal Distribution
    5. One-Sided Confidence Interval
- Chapter 8: Hypothetical Test
    1. Basic Ideas and Concepts of Hypothetical Test
    2. Hypothesis Testing for Population Mean in a Normal Distribution
    3. Hypothesis Testing for Population Variance in a Normal Distribution
    4. Nonparametric Hypothesis Testing

Mode of delivery

Contact teaching

Learning activities and teaching methods

Contact teaching, homework, exam.

Recommended or required reading

(1) Sheng Ju, Xie Shiqian, Panchengyi: Probability & Statistics, 2008, ISBN 978-7-04-023896-9.

(2) Probability and Statistics， Morris H.DeGroot， Addison-Wesley Publishing Company， Second Edition，1989.

Assessment methods and criteria

exercises, course attendance, exam.

Physics Part

2 Courses

College Physics A I (NJIT)

College Physics A II (NJIT)

Name

College Physics A I

NJIT Credits (NJIT)

3 NJIT credits / 4.5 ECTS credits (converted) / 120 hours of work

Language of instruction

Chinese

Timing

1st year, 2nd semester

Learning outcomes

After completing the course, the student will be able to:

- master the basic knowledge of classical mechanics covered by this course and lay the foundation for the study of other subjects in engineering
- develop scientific thinking and methods, noble emotions, and morals, establish a materialistic worldview and a grand cosmology
- apply covered theory to practice and combine basic knowledge of physics with engineering

Contents

College Physics, which is based on the fundamentals of physics, is an important general course for engineering students in higher education. It lays the necessary physics foundation, enhances students' ability to analyze and solve problems, and cultivates students' spirit of exploration and innovation. The basic concepts, theories, and methods taught in this course constitute an important part of students' scientific literacy and are necessary for a scientific worker and engineering technician. This course introduces the basic concepts of classical mechanics and its fundamental rules.

- Chapter 1: Particle Kinematics and Newton's Laws of Motion
- Chapter 2: Law of Conservation of Momentum and Law of Conservation of Energy
- Chapter 3: Motion of Continuous Media
- Chapter 4: Mechanical Vibration
- Chapter 5: Mechanical Wave
- Chapter 6: Wave Optics

Mode of delivery

Contact teaching

Learning activities and teaching methods

Contact teaching, homework, exam.

Recommended or required reading

Liu Yangzheng: Physics and its engineering applications, 2015, ISBN: 9787040416459

Assessment methods and criteria

exercises, course attendance, exam.

Name

College Physics A II

NJIT Credits (NJIT)

3 NJIT credits / 4.5 ECTS credits (converted) / 120 hours of work

Language of instruction

Chinese

Timing

2nd year, 1st semester

Learning outcomes

After completing the course, the student will be able to:

- master the basic knowledge of classical mechanics covered by this course and lay the foundation for the study of other subjects in engineering
- develop scientific thinking and methods, noble emotions, and morals, establish a materialistic worldview and a grand cosmology
- apply covered theory to practice and combine basic knowledge of physics with engineering

Contents

College Physics, which is based on the fundamentals of physics, is an important general course for engineering students in higher education. It lays the necessary physics foundation, enhances students' ability to analyze and solve problems, and cultivates students' spirit of exploration and innovation. The basic concepts, theories, and methods taught in this course constitute an important part of students' scientific literacy and are necessary for a scientific worker and engineering technician. This course introduces the basic concepts of thermodynamic, electromagnetic, and their fundamental rules.

- Chapter 7: Kinetic Theory of Gases
- Chapter 8: Thermodynamic Basis
- Chapter 9: Electrostatic Field
- Chapter 10: Constant Magnetic Field
- Chapter 11: Electromagnetic Induction

Mode of delivery

Contact teaching

Learning activities and teaching methods

Contact teaching, homework, exam.

Recommended or required reading

Liu Yangzheng: Physics and its engineering applications, 2015, ISBN: 9787040416459

Assessment methods and criteria

exercises, course attendance, exam.

Programming Part

5 + 2 Courses

Introduction to Programming (UO)

Programming 2 (UO)

Programming 3 (UO)

Web Development Techniques (NJIT)

Programming 4 (UO)

Software Engineering Project Training (NJIT)

JAVA EE Techniques (NJIT)

Name

Introduction to Programming

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

1st year, 2nd semester

Learning outcomes

After completion of this course, the student will be able to:

- create simple working programs
- identify and use the basic control structures of a program
- identify the concepts of modularity, table, storage of information
- apply the concepts of modular structure, tables, and information storage techniques into a program
- find and fix errors in the program
- solve a computational problem by using abstraction and stepwise refinement
- explain the concept of recursion
- operate with binary and hexadecimal number systems, as well as know the presentation of numbers on a computer
- document the program

Contents

1. Software design method (waterfall) 2. Problem-solving 3. Stepwise refinement 4. Control structures 5. Modular programming, calling modules, communication between modules 6. Data types 7. Arrays 8. Pointers 9. Character strings 10. Data structures 11. Storing data.

Mode of delivery

Contact teaching

Learning activities and teaching methods

Lectures, programming exercises, independent work

Recommended or required reading

Deitel Paul, Deitel Harvey (2007): C How to Program, 8th Edition.

Assessment methods and criteria

Programming assignments and Exams

Grading

Numerical scale 1-5 or fail

Person responsible

Päivi Raulamo-Jurvanen

Name

Programming 2

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

2nd year, 2nd semester

Learning outcomes

After completion of this course, the student will be able to:

- describe the principles of object paradigm (encapsulation, polymorphism, inheritance, composition), generics, and design patterns and can utilize these concepts when creating software,
- describe exception and error management and create fault-tolerant programs,
- explain the connection between the UML models and the source code,
- test an application and interpret the structure and functionality of the source code, as well as
- use basic programming tools, such as a version control system, an IDE, and code analysis tools.

Contents

The concept of an object, encapsulation, composition, inheritance, polymorphism, exceptions, UML charts and code, generics (templates), libraries, containers, design patterns, development tools, version control, documenting, unit testing.

Mode of delivery

Blended teaching

Learning activities and teaching methods

Lectures, laboratory exercises, weekly assignments, independent work

Prerequisites and co-requisites

Introduction to Programming

Recommended or required reading

Timothy Budd (2002): An Introduction to Object-Oriented Programming, 3rd Edition.

Assessment methods and criteria

Programming assignments, weekly assignments, Exam

Grading

Numerical scale 1-5 or fail

Person responsible

Alireza Haghighatkhah

Name

Programming 3

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

3rd year, 1st semester

Learning outcomes

After completion of this course, the student will be able to:

- recognize the influence and requirements of the interface specification on the server development and can apply them in his/her work.
- implement and document a good quality database and use it in an application.
- implement and document the server functionality of a client-server application and apply concurrency when appropriate.
- use existing programming interfaces and message passing protocols in a server application
- test a server application and interpret code written by someone else.
- use programming tools, such as a version control system, an IDE, and code analysis tools in the server development.

Contents

Databases, database programming, data formats, the design, implementation, and testing of a server interface, the safety and security of a server, concurrency.

Mode of delivery

Blended teaching

Learning activities and teaching methods

Lectures, exercises, weekly assignments, independent work

Prerequisites and co-requisites

Programming 2, Data Structures and Algorithms

Recommended or required reading

Announced at the beginning of the course

Assessment methods and criteria

Programming assignments and coursework

Grading

Numerical scale 1-5 or fail

Person responsible

Maëlick Claes

Name

Web Development Techniques

NJIT Credits (NJIT)

3 NJIT credits / 4.5 ECTS credits (converted) / 120 hours of work

Language of instruction

Chinese

Timing

3rd year, 2nd semester

Learning outcomes

After completing the course, the student will be able to:

- understand the basic process of project development
- participate in the project preparation, design, development, testing, delivery, and other different stages of work through the actual project development
- understand the development process of the web project
- master knowledge involved in Java Server Page (JSP) development

Contents

- Chapter 1: Overview of JSP
  1. JSP Technology Overview
  2. JSP Technical Characteristics and Process
  3. Comparison of JSP and Other Server-side Scripting Languages
  4. JSP Development Environment Construction and Tools
  5. JSP Program Development Model
  6. The First JSP Application
- Chapter 2: Basics of JSP Development
  1. Java Language Foundation
  2. JavaScript Scripting Language
- Chapter 3: JSP Syntax
  3. Understand the Basic Structure of JSP
  4. JSP Instruction Mark
  5. JSP Script Identification
  6. JSP Comment and Action Mark
- Chapter 4: JSP Built-in Object
  1. Overview of JSP Built-in Object
  2. Request Object and Response Object
  3. Session Object
  4. Application Object
  5. Out Object
  6. Other Built-in Object
- Chapter 5: JavaBean Technology
  1. Overview of JavaBean
  2. Attributes in JavaBean
  3. Application of JavaBean

4. Application Examples of JavaBean

- Chapter 6: Servlet Technology
  1. Servlet Basics
  2. Servlet API Programming Common Interfaces and Classes
  3. Servlet Development
  4. Servlet Filter
  5. Servlet Listener
  6. Servlet Application Examples
- Chapter 7: JSP Utility Components
  1. JSP File Operation
  2. Send E-mail
  3. JSP Dynamic Chart
  4. JSP Report
- Chapter 8: JSP database Application Development
  1. Database Management System
  2. Overview and Commonly Used Interfaces in JDBC
  3. JDBC Access Database Process
  4. Typical JSP Database Connection
  5. Database Operation Technology
  6. Connection Pool Technology
- Chapter 9: JSP and Ajax
  1. Understand Ajax
  2. Use the XMLHttpRequest Object
  3. Traditional Ajax Workflow
  4. JQuery Implements Ajax
  5. Several Issues that Need Attention in Ajax Development
- Chapter 10: JSP Technology
  1. EL Expression
  2. JSTL Standard Tag Library
  3. Envelopment of Custom Tag Library
  4. SP Framework Technology
- Chapter 11: JSP Comprehensive Development Practice

Mode of delivery

Contact teaching

Learning activities and teaching methods

Contact teaching, experiment, exam.

Recommended or required reading

Jia Zhicheng, Wang Yun, JSP programming design (MOOC version), 2016, ISBN: 9787115417633

Assessment methods and criteria

Experiment report, course attendance, exam.

Name

Programming 4

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

3rd year, 2nd semester

Learning outcomes

After completion of this course, the student will be able to:

- recognize the influence and requirements of the design/implementation interface on the user interface development process and can apply them in his/her work,
- utilize UI libraries and frameworks in his/her application,
- implement and document the client functionality of a client-server application,
- test the application and test and interpret the code and the application structure with its effects on testing, maintenance, and further development,
- use programming tools, such as a version control system, an IDE, and code analysis tools, as well as
- act as a member of a software development team.

Contents

User interface elements, foundations of user interface libraries, user interface design principles, user interface layout, the relationship between user interfaces and software architectures, web usability, web user interfaces, web programming.

Mode of delivery

Blended teaching

Learning activities and teaching methods

Lectures, exercises, coursework, independent work

Prerequisites and co-requisites

Programming 3

Recommended or required reading

Lauesen Soren (2005). User Interface Design: A Software Engineering Perspective.

Assessment methods and criteria

Assignments, exams, coursework

Grading

Numerical scale 1-5 or fail

Person responsible

Alireza Haghighatkhah

Name

Software Engineering Project Training

NJIT Credits (NJIT)

4 NJIT credits / 6 ECTS credits (converted) / 160 hours of work

Language of instruction

Chinese

Timing

4th year, 1st semester

Learning outcomes

After completing the course, the student will be able to:

- understand the development processes by participating in different stages such as project preparation, design, development, testing, and delivery of real project cases in enterprises.
- write software development documents and specifications.

Contents

- MVVM architecture pattern
- REST API backend development
- Mobile application development on Harmony OS
- WeChat mini program development

Mode of delivery

Contact teaching

Learning activities and teaching methods

Contact teaching, homework, course project.

Assessment methods and criteria

course attendance, homework, project report.

Name

JAVA EE Techniques

NJIT Credits (NJIT)

3 NJIT credits / 4.5 ECTS credits (converted) / 120 hours of work

Language of instruction

Chinese

Timing

4th year, 1st semester

Learning outcomes

After completing the course, the student will be able to:

- understand advanced web application development concepts and methodology,
- develop enterprise-level web applications using the SSM (Spring + Spring MVC + Mybatis).

Contents

- Aspect-oriented programming
- Dependency injection and inversion of control
- MVC architecture pattern
- Object relational mapping
- Web application development using SSM

Mode of delivery

Contact teaching

Learning activities and teaching methods

Contact teaching, experiment, homework, exam.

Assessment methods and criteria

Experiment report, course attendance, homework, exam.

Computer Systems Architecture Part

4 Courses

Databases (UO)

Computer Network A (NJIT Transferred)

Data Modeling and Design (UO)

Operating System A (NJIT Transferred)

Name

Databases

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

2nd year, 1st semester

Learning outcomes

After completing the course, the student will be able to:

- apply the theory of the relational databases and the basics of set theory,
- build a good quality relational database and use queries,
- use a relational database for storing persistent objects,
- use conceptual modeling for designing databases, as well as
- normalize a database and assess its quality.

Contents

- Conceptual modeling
- Relational model and database
- SQL
- Quality of database
- Storing objects to the relational database

Mode of delivery

Contact teaching

Learning activities and teaching methods

Lectures, exercises, independent work

Prerequisites and co-requisites

-

Recommended or required reading

Coronel C & Morris S (2018), Database systems: design, implementation, and management.

Assessment methods and criteria

Assignments, Exams

Grading

Numerical scale 1-5 or fail

Person responsible

Alireza Haghighatkhah

Name

Computer Network A

ECTS Credits (NJIT Transferred)

5 ECTS credits / 133 hours of work

Language of instruction

Chinese

Timing

2nd year, 2nd semester

Learning outcomes

After completing the course, the student will be able to:

- master the fundamentals of the computer network; Be familiar with the composition and functional classification of the computer network; Master the performance indexes and non-performance characteristics of the computer network; Understand the topology of the computer network.
- understand the concept of data communication, information coding technology, and communication mode; Understand transmission media, multiplexing technology, and bandwidth access technology.
- master the architecture of the computer network (OSI and TCP / IP architecture); Be familiar with the functions and data units of different layers.
- understand wireless local area network (WLAN) and mobile network; Understand wireless personal area network (WPAN), wireless metropolitan area network (WMAN), and cellular mobile communication network technology.
- understand the meaning of address of each network layer; Be familiar with the application of physical address and logical address in the network and be proficient in conversion and calculation; Understand the concept and structure of domain name system (DNS).
- understand the related concepts, characteristics, and various security policies of network security.
- understand the related concepts of WAN technology; Understand the concept and working principle of point-to-point protocol (PPP); Understand the application of wireless network in WAN.

Contents

- Chapter 1: Computer Network Fundamental
  1. The Definition and Development of Computer Network
  2. The Composition of Computer Network
  3. The Function and Classification of Computer Network
  4. Computer Network Topology
- Chapter 2: Data Communication Fundamental
  1. Basic Concepts of Data Communication
  2. Information Coding Technology
  3. Data Transmission and Communication Method
  4. Transmission Media
  5. Multiplexing Technology
  6. Data Switching Technology
  7. Error Control
- Chapter 3: Computer Network Architecture
  1. Network Architecture and Network Protocol

2. Physical Layer
3. Data Link Layer
4. Network Layer
5. Transport Layer
6. Session Layer
7. Presentation Layer
8. Application Layer

- Chapter 4: Local Area Network (LAN) Technology
   1. Local Area Network (LAN) Fundamental
   2. Local Area Network (LAN) Hierarchical Structure and Standardization Model
   3. IEEE802.3 Standard, IEEE802.4 Standard, IEEE802.5 Standard
   4. High-speed LAN and Wireless Local Area Network (WLAN)
- Chapter 5: TCP/IP Protocol
   1. TCP/IP Fundamental
   2. Network Layer Protocol
   3. Transport Layer Protocol
   4. Application Layer Protocol
   5. Web Application Programming Interface
- Chapter 6: Network Equipment and Network Interconnection
   1. Network Interconnection Fundamental
   2. Network Interconnection Equipment
   3. Basic Configuration Operation of Network Equipment
   4. Switch Address and Router Management
   5. Network Interconnection
- Chapter 7: Network Technology
   1. VLAN Technology
   2. Redundant Link
   3. Network Address Translation
   4. Access Control List
   5. Network Communication Detection Tool
- Chapter 8: Wide Area Network (WAN) Technology
   1. Wide Area Network (WAN) Fundamental
   2. Main Features of WAN
   3. WAN Data Exchange and WAN Protocol
   4. Public Communication Network
   5. Windows Common Network Commands

Mode of delivery

Online teaching

Learning activities and teaching methods

Online teaching, homework, experiment, exam.

Recommended or required reading

Xie Xiren: Computer Network, 2017, ISBN: 978-7-121-30295-4

James F. Kurose Keith W. Ross: Computer Networking: A Top-Down Approach, 4th Edition, ISBN: 9787111599715

Andrew S. Tanenbaum: Computer network (4th Edition), translated by Pan Aimin, Tsinghua University Press. ISBN: 9780130661029.

Assessment methods and criteria

Experiment report, course attendance, exam.

Name

Data Modeling and Design

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

3rd year, 1st semester

Learning outcomes

After completing the course, the student will be able to:

- compare traditional relational databases to modern distributed databases, as well consider the influence of CAP-theorem to distributed databases,
- identify features affecting the quality of non-relational databases and choose appropriate implementation of a non-relational database for use,
- explain data persistency concepts and can apply database transaction management principles while using database systems,
- describe (typical) contemporary database solutions and their role in large-scale software systems (such as ERP).

Contents

Modern database solutions and the use of them as well transactions, concurrency, and recovery.

Mode of delivery

Blended teaching

Learning activities and teaching methods

Lectures, exercises, independent work

Prerequisites and co-requisites

Databases, Software Quality and Testing,

Recommended or required reading

Ramez Elmasri, Shamkant Navathe (2015). Fundamentals of database systems, 7th Edition.

Assessment methods and criteria

Assignments, exams

Grading

Numerical scale 1-5 or fail

Person responsible

Nirnaya Tripathi

Name

Operating System A

ECTS Credits (NJIT Transferred)

5 ECTS credits / 133 hours of work

Language of instruction

Chinese

Timing

3rd year, 1st semester

Learning outcomes

After completing the course, the student will be able to:

- master the objective and function, development process, the basic concepts, and main functions of the operating system
- understand the structural design of the operating system
- master the concept of process, the concept of concurrent process, the concept of process synchronization, the basic method of achieving process synchronization with a mutual exclusion, and process communication
- understand the concept of thread
- understand causes and necessary conditions of deadlock, and methods to prevent deadlocks
- master the basic concept of the memory system and memory management
- master various memory management methods and page replacement algorithm
- master the four I/O control methods, the reason for buffer introduction, device allocation, dish scheduling algorithm
- understand I/O system, device processor, and disk memory management
- understand the concepts of files and the file system, the logical and physical structures of files, directory management methods
- understand file memory space management, file sharing and protection, and data consistency control
- understand the Windows operating system and the Linux operating system

Contents

- Chapter 1: Operating System Fundamental
  1. Operating System Concept
  2. Operating System Functions
  3. Operating System Interface
  4. Operating System Structure
- Chapter 2: Processor Management and Concurrent Processes
  1. Multiprogramming Design
  2. Process
  3. Process Control
  4. Process Scheduling
  5. Thread and Thread Implementation
  6. The Concept of Concurrent Process
  7. Mutual Exclusion and Synchronization of Process
  8. Tube

9. Process Communication
10. Deadlock

- Chapter 3: Memory Management
  1. Memory System Fundamental
  2. Memory Management Fundamental
  3. Partition Memory Management
  4. Simple Paged Memory Management
  5. Simple Segmented Memory Management
  6. Virtual Memory Management
  7. Request Paged Virtual Memory Management
  8. Request Segmented Memory Management
  9. Request Segment Page Virtual Memory Management
- Chapter 4: Device Management
  1. Device Management Fundamental
  2. Device I/O Control Method
  3. Device I/O Software Principle
  4. Buffer Technology
  5. Peripheral Device Allocation, Recycling, and Start
  6. Disk Drive Scheduling
  7. Virtual Device
- Chapter 5: File Management
  1. File System
  2. File Directory
  3. File Structure and Access Method
  4. File Access and Sharing
  5. Security and Protection
- Chapter 6: Window and Linux Operating System
  1. Windows 2000/XP Operating System
  2. Linux Operating System

Mode of delivery

Contact teaching

Learning activities and teaching methods

Contact teaching, homework, exam.

Recommended or required reading

Tu Lizhong, Xu Jinbao, instruction for operating system 2013, ISBN: 978-7-121-20509-5.

Assessment methods and criteria

Exercises, course attendance, exam.

Theoretical Computer Science Part

2 Courses

Data Structures and Algorithms (UO)

Principles of Compiling (NJIT Transferred)

Name

Data Structures and Algorithms

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

2nd year, 2nd semester

Learning outcomes

After completing the course, the student will be able to:

- select data structures and algorithms to an application,
- apply induction when proving algorithm correctness, and define recursive algorithms,
- describe trees, graphs, and their basic algorithms and apply them in a program,
- describe the most common sorting algorithms,
- analyze the correctness and time complexity of an algorithm implemented in a program.

Contents

- Basic data structures
- Analysis of algorithms
- Sorting algorithms
- Hash tables
- Binary search trees
- Graphs and their algorithms
- Algorithm design paradigms

Mode of delivery

Blende teaching

Learning activities and teaching methods

Lectures, exercises, independent work

Prerequisites and co-requisites

Programming 2, Databases

Recommended or required reading

Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to algorithms, 3rd Edition.

Assessment methods and criteria

Assignments, Exams

Grading

Numerical scale 1-5 or fail

Person responsible

Pertti Karhapää

Name

Principles of Compiling

ECTS Credits (NJIT Transferred)

5 ECTS credits / 133 hours of work

Language of instruction

Chinese

Timing

3rd year, 1st semester

Learning outcomes

After completing the course, the student will be able to:

- understand the structure and workflow of the compiling system, the design principles, and implementation techniques of each component of the compiling system; Gain the preliminary ability to analyze, design, implement and maintain the compiling system.
- deeply understand the programming language and program execution process; Improve the overall understanding of the computer system.
- understand computing from the perspective of language translation and representation transformation; Know how to apply different computational thinking methods (such as abstraction, automation, recursion, decomposition, and trade-offs in the compilation process) in solving practical problems like natural language processing, program verification, and network information processing.

Contents

This course mainly contains the basic principles, methods, techniques of the construction, design, and implementation of the compiler. Develop students' ability to analyze and solve problems, conduct system analysis, and cultivate simple systems. Lay a solid foundation for further study of computer science and future work such as software development.

- Chapter 1: General Remarks
  1. Programming Language and Program
  2. Compiler Construction and Related Concepts
  3. Formal Language Theory and Compiler Implementation Techniques
- Chapter 2: Grammar and Language
  1. Symbol String and Set of Symbol Strings
  2. The Formal Definition of Grammar and Language
  3. Language Classification
  4. Grammatical Equivalence and Equivalent Exchange
  5. Parse Trees and Sentence Parsing
- Chapter 3: Lexical Analysis
  1. Regular Expressions and Finite Automaton
  2. Implementation of Lexical Analyzer
  3. Automatic Generation of Lexical Analyzer
- Chapter 4: Parsing - Top-down Parsing Technique
  1. Top-down Parsing Techniques with Backtracking
  2. Top-down Parsing Techniques without Backtracking

- Chapter 5: Parsing - Bottom-up Parsing Technique
  1. Operator-first Parsing Technique
  2. LR(k) Parsing Technique
  3. The Implementation of Sentence Pattern Parsing of LR(1) Recognition Program
- Chapter 6: Semantic Analysis and Target Code Generation
  1. Explanation of the Translation
  2. Object Code Generation
  3. Consideration in Implementation of Semantic Analysis
  4. The Intermediate Representation Code of the Source Program
- Chapter 7: Operating Environment
  1. Storage Allocation Strategy
  2. Register Allocation
  3. Symbol Table
- Chapter 8: Code Optimization
  1. Basic Block and Flow Graph
  2. Optimization of Basic Block
  3. Optimizations Related to Loops
  4. Peephole Optimization
- Chapter 9: Checking and Correcting Program Errors
  1. Recovery and Correction of Lexical Error
  2. Recovery and Correction of Grammatical Error
  3. Semantic Error

Mode of delivery

Contact teaching

Learning activities and teaching methods

Contact teaching, homework, exam.

Recommended or required reading

Zhang Xiner, Computer’s principles of compiling (third edition) 2008, ISBN: 978-7-03-021247-4.

Assessment methods and criteria

Exercises, course attendance, exam.

Other Part – Related to Computer Science

11 + 1 Courses

Enterprise Perceptual Practice (NJIT Transferred)

Introduction to Software Engineering (UO)

Information Systems in Organizations (UO)

Requirements Engineering (UO)

Information Security (UO)

Software Quality and Testing (UO)

Software Modeling and Design (UO)

Big data and its application (NJIT Transferred)

Software Architectures (UO)

Software Project Management (NJIT)

AI and Software Engineering (UO)

Special Course in Software Engineering (UO)

Name

Enterprise Perceptual Practice

ECTS Credits (NJIT Transferred)

1 ECTS credits / 27 hours of work

Language of instruction

English

Timing

1st year, 1st semester

Learning outcomes

After completing the course, the student will be able to:

- understand the role of software engineering in companies,
- understand how to utilize the concepts of software engineering to improve the work efficiency

Contents

1. Visit IT companies 2. Introduction to UO students about how to utilize the knowledge of software engineering within the context of application

Mode of delivery

Contact teaching

Learning activities and teaching methods

Visiting IT companies, independent work

Prerequisites and co-requisites

-

Recommended or required reading

-

Assessment methods and criteria

Active participation and accepted assignments

Grading

Numerical scale 1-5 or fail

Person responsible

Jouni Markkula

Name

Introduction to Software Engineering

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

1st year, 2nd semester

Learning outcomes

After completing the course, the student will be able to:

- describe the principles, define the key concepts and use professional terminology of software engineering,
- describe software engineering as a professional practice and a field of industry,
- describe and work following professional practices that are important for software engineers,
- describe contemporary software processes and choose appropriate ones for specific situations,
- describe and apply valid problem identification and structuring methods in software engineering,
- identify and apply some contemporary software engineering models, methods, and tools, as well as
- describe the necessity of continuous learning and professional development.

Contents

- Principles of professional software development
- Software processes
- Agile software development
- Requirements engineering
- System modeling
- Architectural design
- Design and implementation
- Software testing
- Software evolution

Mode of delivery

Contact teaching

Learning activities and teaching methods

Lectures, group exercises, independent work

Recommended or required reading

Sommerville, Ian (2016). Software Engineering, 10th Edition.

Assessment methods and criteria

Moodle exercises and essays

Grading

Numerical scale 1-5 or fail

Person responsible

Markku Oivo

Name

Information Systems in Organizations

ECTS Credits (UO)

The credits of the course in this version are 5 ECTS, while it is 6 ECTS in my transcript.

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

1st year, 2nd semester

Learning outcomes

After completing the course, the student will be able to:

- explain the importance of information systems in organizations
- define the conditions for the successful operation of the information in the organization
- explain the main features of the development of information systems.

Contents

The basics issues of organization, structure, and operation, the basics of a digital organization, information types, and roles of the organizations, interaction between information and organization, the role of information systems in the management of organizations and decision-making, formation of organizational knowledge and management, enterprise resource planning (ERP) systems, organizational reform of information systems and the economic importance of information systems.

Mode of delivery

Contact teaching

Learning activities and teaching methods

Lectures, weekly assignments, essays, independent work

Prerequisites and co-requisites

-

Recommended or required reading

Wallace Patricia: Information Systems in Organizations - People, Technology, and Processes.

Laudon Kenneth C., Laudon Jane P. Management Information Systems - Managing the Digital Firm,14th edition.

Assessment methods and criteria

Active participation and assignments

Grading

Numerical scale 1-5 or fail

Person responsible

Tero Päivärinta

Name

Requirements Engineering

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

2nd year, 1st semester

Learning outcomes

After completing the course, the student will be able to:

- apply requirements engineering skills and techniques individually and in teams, and understands the fundamentals of the requirements,
- choose and apply some of the requirements elicitation techniques,
- choose and apply some of requirements specification and documentation techniques, as well as
- apply appropriate requirements validation techniques, as well as learn new requirements engineering methods and techniques.

Contents

- Requirements traceability
- Different stakeholder viewpoints and requirement categories
- Requirements change
- Problem structuring methods
- Requirements engineering skills and techniques in the iterative development environment
- Requirements identification, elicitation, specification, and documentation techniques
- Requirements prioritization and validation techniques

Mode of delivery

Contact teaching

Learning activities and teaching methods

Lectures, exercises, group work, independent work

Prerequisites and co-requisites

Introduction to Software Engineering

Recommended or required reading

Wiegers, Karl & Beatty, Joy (2013). Software Requirements, 3rd Edition.

Assessment methods and criteria

Active participation, assignments, coursework

Grading

Numerical scale 1-5 or fail

Person responsible

Pertti Seppänen

Name

Information Security

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

2nd year, 1st semester

Learning outcomes

After completing the course, the student will be able to:

- define essential information security concepts and components of information systems security,
- recognize the common types of security threats and their managerial and technical protection mechanisms,
- describe the tasks and responsibilities of information security professionals,
- explain the different phases of secure systems development/acquisition,
- recognize the fundamental characteristics of risk management and evaluate information security risks,
- recognize basics of technical information security methods and cryptography, as well as
- explain areas of behavioral information security research and their practical implications.

Contents

- Basic concepts of information security
- Information security threats, vulnerabilities, and risks
- Legal issues and information security frameworks
- Risk management
- Cryptography
- Information security technologies
- Behavioral information security research

Mode of delivery

Contact teaching

Learning activities and teaching methods

Lectures, weekly assignments, essays

Prerequisites and co-requisites

-

Recommended or required reading

Whitman, M.E. Mattord, H.J.: Principles of Information Security, 5th Edition.

Assessment methods and criteria

Assignments, essays

Grading

Numerical scale 1-5 or fail

Person responsible

Elina Annanperä

Name

Software Quality and Testing

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

2nd year, 2nd semester

Learning outcomes

After passing the course, the student will be able to:

- describe different views on software quality and the role of testing in software engineering,
- detect defects in software using different techniques,
- describe testing levels and techniques,
- create test cases and conduct unit testing with appropriate testing tools,
- describe the basics of test-driven development and test automation, as well as
- define the scope of software testing and quality assurance projects.

Contents

1. Why Testing and Software quality is important
2. Testing as a process
3. Testing as a technique
4. Designing tests (using testing techniques and domain knowledge)
5. Oracles and Coverage
6. Unit testing and TDD

Mode of delivery

Blended teaching

Learning activities and teaching methods

Lectures, exercises, group work, independent work

Prerequisites and co-requisites

Software Modeling and Design

Recommended or required reading

Announced at the beginning of the course

Assessment methods and criteria

Assignments, exams, coursework

Grading

Numerical scale 1-5 or fail

Person responsible

Pertti Seppänen

Name

Software Modeling and Design

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

2nd year, 2nd semester

Learning outcomes

After completing the course, the student will be able to:

- apply possibilities of UML-language family to describe different views of software development,
- produce detailed descriptions using static and dynamic modeling and code from design models,
- describe principles of object-orientation, reverse engineering, and general principles and concepts of software design (such as abstraction, modularization, cohesion, connectedness), as well as
- describe the role of best practices in software modeling and design.

Contents

UML notation and methodology. UML -diagrams (class, sequence, and static diagrams, etc.). Principles of object orientation and quality aspects of it. Design best practices.

Mode of delivery

Blended teaching

Learning activities and teaching methods

Lectures, exercises, independent work

Prerequisites and co-requisites

Programming 2, Requirements Engineering

Recommended or required reading

Simon Bennett, Steve McRobb & Ray Farmer: Object-oriented systems analysis and design using UML, 4th Edition.

Assessment methods and criteria

Assignments, exams

Grading

Numerical scale 1-5 or fail

Person responsible

Päivi Raulamo-Jurvanen

Name

Big data and its application

ECTS Credits (NJIT Transferred)

5 ECTS credits / 133 hours of work

Language of instruction

Chinese

Timing

3rd year, 1st semester

Learning outcomes

After completing the course, the student will be able to:

- master the collection, transmission, processing, application, and other techniques of big data
- understand the infrastructure of the big data analysis system
- master the corresponding storage and analysis techniques
- understand HBase, Hive, Spark, and other related big data technologies and be able to build big data platforms with practical engineering applications

Contents

- Chapter 1: Big Data Foundation

This section introduces the basic concepts, impacts, and application areas of Big Data, and explains the relationship between Big Data, Cloud Computing, and the Internet of Things. Since Hadoop has become the most widely used Big Data technology, this section mainly introduces the Hadoop Big Data processing architecture, including Hadoop MapReduce, HDFS, HBase, etc.

This section includes two chapters:

1. Concept and Application of Big Data and Analysis of the Relationships Among Big Data, Cloud Computing, and the Internet of Things
2. Big Data Processing Architecture of Hadoop

- Chapter 2: Big Data Storage

This section introduces the concepts and principles of big data storage-related technologies, including distributed file system HDFS, distributed database HBase, NoSQL database, and cloud database. HDFS provides the capability of large-scale distributed file storage in a cluster of inexpensive servers. HBase is a highly reliable, high-performance, column-oriented and scalable distributed database, mainly used for storing NoSQL database can support ultra-large-scale data storage with solid horizontal scalability, which can effectively make up for the shortage of traditional relational databases. NoSQL's flexible data model can well support Web2.0 applications. Cloud database is deployed and virtualized in the cloud computing environment. It can free users from cumbersome database hardware customization and, at the same time, allow users to have powerful database scalability to meet the data storage needs of various users.

This section includes four chapters:

1. Hadoop Distributed File System (HDFS)
2. Distributed Database - HBase
3. NoSQL Database
4. Introduction to Cloud Database

- Chapter 3: Big Data Processing and Analysis

This section introduces the techniques related to Big Data processing and analysis. Big data includes static data and dynamic data (streaming data). Static data is suitable for batch processing, while dynamic data requires real-time computation. MapReduce, a distributed parallel programming framework, can significantly improve program performance and enable efficient batch data processing. spark, a memory-based distributed computing framework, is a fast and versatile engine applied to large-scale data processing. Streaming computing framework Storm is a low-latency, scalable, and highly reliable processing engine that can be used to solve real-time computing problems for streaming data. Pregel is a new graph computing framework for solving large-scale graph computing problems. In addition, data visualization is the last part of big data analysis, and this paper will briefly introduce the concept of data visualization and related tools.

This part includes six chapters:

1. Distributed Parallel Programming Framework - MapReduce
2. Further Discussion of Hadoop
3. Memory Based Distributed Computing Framework - Spark
4. Open-source Stream Computing Framework - Storm
5. Graphs Calculation Framework - Pregel
6. Concept of Data Visualization and Related Tools

- Chapter 4: Big Data Application

Big data has been widely used in social production and daily life and has played an important role in promoting the development and progress of human society. This article introduces the application of big data on the Internet, biomedicine, logistics, urban management, finance, automobile, retail, catering, telecommunications, energy, sports and entertainment, security, government, daily life, etc. Students can deeply feel the impact of big data on society and its essential value by studying this section

This section includes three chapters:

1. Application of Big Data in The Internet Field with the Recommendation System as the Core
2. Application of Big Data in Biomedicine
3. Application of Big Data in Other Fields

Mode of delivery

Contact teaching

Learning activities and teaching methods

Contact teaching, experiment, exam.

Recommended or required reading

Li Ziyu, Principles and application of big data technology （2nd），2021, ISBN: 9787115443304.

Assessment methods and criteria

Experiment report, course attendance, assignment.

Name

Software Architectures

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

3rd year, 2nd semester

Learning outcomes

After passing the course, the student is able to:

- describe the concepts and techniques of the software architecture design – especially in the case of object-oriented design,
- describe typical architecture solutions of main-stream modern software solutions – for instance, apps of smart devices and server-based systems,
- identify and analyze the pros and cons of different software architectures from the viewpoints of software design & implementation, software execution, software quality, and software maintainability,
- use UML modeling techniques to describe different perspectives of software architecture,
- create different optional architectural solutions for software based on its functional and non-functional requirements and evaluate the applicability of the optional architectures to the problem in question,
- describe the role of architectural design in agile and iterative software development processes.

Contents

The fundamentals of software architectures. Documenting software architectures. Components and interfaces, Software dependencies. Design patterns. Architectural styles. Evaluation methods of software architectures. Agile and iterative software development processes and software architecture design.

Mode of delivery

Blended teaching

Learning activities and teaching methods

Lectures, exercises, group work

Prerequisites and co-requisites

Data Modeling and Design

Recommended or required reading

Hanmer Robert (2013). Pattern-oriented Software Architecture for Dummies, 2nd Edition. Bass L., Clements R., Kazman R.: Software Architecture in Practice 3rd Edition. Agile Software Architecture 1st Edition. Aligning Agile Processes and Software Architectures.

Assessment methods and criteria

Exercises, assignments, exams

Grading

Numerical scale 1-5 or fail

Person responsible

Maëlick Claes

Name

Software Project Management

NJIT Credits (NJIT)

2 NJIT credits / 3 ECTS credits (converted) / 80 hours of work

Responsible Department

School of Computer Engineering

Language of instruction

Chinese

Timing

3rd year, 2nd semester

Learning outcomes

After completing the course, the student will be able to:

- master the characteristics and general process of software project management
- master the work content in each stage of software project management
- understand the misunderstandings and constraints in software project management
- master the basic principles and effective methods that should be followed in software project management
- master the writing methods of important technical documents in software project management

Contents

“Software Project Management” enables students to understand software project management, master the basic methods of software project management, and have a comprehensive understanding of the initiation of software projects, project plans, execution control of software projects, and the end of software. Students need to master the practice and implementation of software project management. Students are trained from the perspective of software engineering to analyze and solve problems by software project management approaches. The course cultivates students' cognition of software projects, software design, development, and maintenance.

- Chapter 1: Overview of Software Project Management
  1. Concept and Characteristics of Project
  2. Software Project and its Characteristics
  3. Concept of Project Management
  4. Project Management Body of Knowledge (based on the ten knowledge fields of PMBOK6 released in 2017)
  5. Bodies of Knowledge Suitable for Software Project Management
- Chapter 2: Project Establishment
  1. Project Evaluation
  2. Project Approval
  3. Project Bidding
  4. Project Charter (Main Content and Functions)
- Chapter 3: Life Cycle Model (Understand the advantages and disadvantages of each life cycle model and applications)
  1. Overview of Life Cycle
  2. Five Life Cycle Models (Predictive, Iterative, Incremental, Agile, and Hybrid Life Cycle Model)

- Chapter 4: Software Project Scope Management – Requirement
  1. Concept and Level of Software Requirement
  2. Composition of Requirement Engineering (requirement elicitation and its main activities, main content of requirement analysis, solution of unclear needs, how to manage requirement changes)
- Chapter 5: Software Project Scope Management –Work Breakdown
  1. Definition of Work Breakdown Schedule (WBS)
  2. Functions, Characteristics, and Decomposition of WBS
- Chapter 6: Software Project Cost Management
  1. Overall Process of Software Project Cost Management
  2. Composition of Software Project Cost
  3. Basic Methods of Software Project Scale Estimation (LoC, FP, UCP)
  4. Basic Methods (expert judgment, analogy, top-down, bottom-up, algorithm model) and Characteristics of Software Project Cost Estimation
  5. Task Decomposition and Function (Understand software project cost budget, Grasp software project cost control and monitoring, understand Project Management Metrics)
- Chapter 7: Software Project Schedule Management
  1. Overall Process of Software Project Schedule Management
  2. Logical Relationships (FS, SS, FF, SF) and Dependencies (mandatory, selective, internal, external) between Activities
  3. Network Diagram of Activity Sequence (Know differences and characteristics of Precedence Diagramming Method (PDM) and Arrow Diagramming Method (ADM))
  4. Methods of Project Duration Estimation (Understand Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM))
  5. Crash in Time Compression Method (Grasp the linear relationship between time compression and cost increase)
  6. Basic Characteristics of Gantt Chart and Milestone Chart
  7. Definition of Forward Method (Grasp basic steps and calculation formula of calculating task duration)
- Chapter 8: Software Project Quality Management
  1. Basic Concept of Software Quality
  2. Software Quality Model (Boehm’s Quality Model and McCall’s Quality Model)
  3. Common Activities of Software Quality Assurance (QA), Quality Control (QC), and Their Differences
- Chapter 9: Software Project Configuration Management
  1. Basic Activities of Software Project Configuration Management
  2. Basic Concept and Purposes of Configuration Management (Grasp software configuration item (SCI), baseline, version, and software configuration control board (SCCB))
- Chapter 10: Software Project Team Management
  1. Classification of Project Organizational Structure (Function, Project, Matrix) (Understand the advantages, disadvantages, and applicability of various types)
  2. Roles and Responsibilities Required by A Software Project Team
  3. Main content of Project Communication Plan
  4. Four Steps of Project Stakeholder Analysis (Distinguish project stakeholder, analyze based on importance, and support, draw project stakeholder analysis grid)
- Chapter 11: Software Project Risk Management
  1. Concept of Risk

2. Source of Software Risk
3. Overall Process of Software Project Risk Management
4. Basic Methods of Risk Identification
5. Basic Methods of Risk Assessment (Analysis)

Mode of delivery

Contact teaching

Learning activities and teaching methods

Contact teaching, experiment, exam.

Recommended or required reading

Han Wanjiang, Jiang Lixin: Software project management A Case Study Approach, 2019, ISBN: 978-7-111-62920-7.

Assessment methods and criteria

course attendance, Assignment.

Name

AI and Software Engineering

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

3rd year, 2nd semester

Learning outcomes

After completing the course, the students will be able to:

- reflect and critically assess the role of AI in software engineering,
- reflect and discuss issues related to the development of intelligent and adaptive systems, as well as
- apply methods and theory learned throughout the course.

Contents

The course consists of three main modules:

- Introduction to the course
- Basics of AI in software-intensive services
- Scenarios of how AI can be applied to solve SE problems

Mode of delivery

Blended teaching

Learning activities and teaching methods

Lectures, assignments, independent work

Prerequisites and co-requisites

Data Modeling and Design

Recommended or required reading

Menzies Tim, Williams Laurie, Zimmermann Thomas (2016). Perspectives on Data Science for Software Engineering. Menzies Tim, Kocaguneli Ekrem, Turhan Burak, Minku Leandro, Peters Fayola (2014). Sharing Data and Models in Software Engineering.

Assessment methods and criteria

Assignments, exams

Grading

Numerical scale 1-5 or fail

Person responsible

Burak Turhan

Name

Special Course in Software Engineering

ECTS Credits (UO)

5 ECTS credits / 133 hours of work

Language of instruction

English

Timing

4th year, 1st semester

Learning outcomes

After completing the course, the students will be able to:

- understand what is a chatbot, the underlying technologies of a chatbot, and how to design and implement a chatbot.
- apply software engineering processes and practices in a team development context.
- develop chatbot applications and integrate them into major or custom platforms.

Contents

1. Introduction to chatbot and application examples. 2. User input and natural language processing. 3. Conversation flow. 4. Designing chatbot personality. 5. Chatbot testing

Mode of delivery

Blended teaching

Learning activities and teaching methods

Lectures, exercise, independent and group work

Prerequisites and co-requisites

-

Recommended or required reading

Eleni Adamopoulou, Lefteris Moussiades (2020). An Overview of Chatbot Technology

Elayne Ruane, Sinead Farrell, Anthony Ventresque (2021). User Perception of Text-Based Chatbot Personality

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, et al (2020). Recipes for building an open-domain chatbot

Assessment methods and criteria

Assignments, quiz, group reports, exams

Grading

Numerical scale 1-5 or fail

Person responsible

Minna Isomursu