SHORT SYLLABI OF THE BSC STUDY PROGRAM IN CSE (2024 – 2027)

YEAR 1

SEMESTER 1

Course title: Linear Algebra with Calculus I (Mandatory, Sem. I, 7 ECTS)
Lecturer: Qefsere Gjonbalaj, Shqipe Lohaj, Valdete Rexhëbeqaj Hamiti

Short course description: In this course will be studied: complex numbers, parts from linear algebra and functions with one-variable.
Course objectives: Students should be trained so that the knowledge gained through this course can be applied as an ancillary device in electrical and computer studies.
Learning outcomes:
Upon completion of this course the student will be able to:

1. solves and formulate various problems in the field of his profession when dealing with complex number operations;
2. describe and solve problems related to systems of linear equations, through matrices and determinants;
3. ascertains the functional connections in research of various electrical phenomena, and then using differential calculations describe and examine those functional connections;
4. understand the concept of the derivative as well as its application in the calculation of different measures in different fields of engineering.

Teaching methodology: Lectures, discussions, exercises, consultations, homework, seminar work, intermediate evaluations, final exam.
Evaluation methods: Attendance and classroom activity (10 %), Seminar work and homework (10%), first intermediate evaluation (25%), second intermediate evaluation (25%), final exam (30%).
Concretization tools/IT: Pencil, whiteboard, projector and computer.
Ratio between theoretical and practical part: Ratio between the theoretical and practical part is 2:1.
Literature:

1. Hamiti E. - Matematika I, Prishtinë 2008.
2. Hamiti E. - Matematika II, Prishtinë 2008.
3. Peci H, Doko M. - Përmbledhje detyrash të zgjidhura nga Matematika I, Prishtinë 1997.
4. Loshaj Z. – Përmbledhje detyrash të zgjidhura nga Matematika II, Prishtinë 1996.

Course title: Physics for Engineering 1 (Mandatory, Sem. I, 6 ECTS)
Lecturer: Dr. Sc. Valon Veliu

Short course description: The course includes basic knowledge of physics (basic principles of mechanics and thermodynamics) necessary to gain general knowledge that is basic in engineering.
Course objectives: Enabling students to use the laws of physics to solve various problems in engineering.

Learning outcomes: After successfully completing this course, the student will be able to:

- analyze and apply problems from mechanics and solve equations of motion;
- analyze and apply problems from translational motion dynamics, Newton's equations;
- analyze and apply the principles of conservation of energy and momentum;
- analyze and apply the equation of circular motion for rigid bodies;
- analyze and apply problems from gravitation, Newton's law of gravity, Kepler's laws;
- analyze and apply problems from fluid mechanics, continuity equation, Bernoulli's equation;
- analyze and apply problems from molecular physics, the law of ideal gasses, the pressure of molecules on the walls of the container;
- analyze and apply the laws of thermodynamics and cyclic processes of thermodynamics.

Teaching methodology: Lectures, numerical exercises, laboratory exercises, seminar, homework, consultations.

Evaluation methods: Classroom activity 5%, Laboratory exercises 7,5%, Homework 7,5%, Seminar work 10%, First periodic evaluation 35 %, Second periodical evaluation 35% or Final Exam 70 %.

Concretization tools/IT: Pencil, whiteboard, projector and computer.

Literature:

1. S. Skenderi, R. Maliqi, "Fizika për studentët e fakulteteve teknike", UP, Prishtinë, 2005.
2. David Halliday, Robert Resnick, Jearl Walker, Fundamentals of Physics Extended, 10th edition, Wiley (2013).
3. Raymond Serway and John Jewett, Physics for Scientists and Engineers with modern physics, 10th edition, Cengage Learning, (2018)
4. James S Walker, - Physics, 5th edition, Pearson Addison-Wesley (2016).

Course title: Fundamentals of electrical engineering 1, (Mandatory, Sem. I, 6 ECTS)
Lecturer: Prof. Dr. Enver Hamiti, Prof. Dr. Mimoza Ibrani, Prof. Ass. Dr. Vjosa Shatri

Short Course description: Basics of electricity. Fundamental lows of electricity. Electric potential. An electric dipole and flux lines. Electrostatic induction. Electric generator. Polarization in dielectrics. Boundary conditions. Capacitance. Electrostatic networks. Energy and forces in electrostatic fields. Electrostatic field analysis using MATLAB. Concepts, elements and topology of electric circuits. Types of electrical circuits. Elementary DC circuits. Circuit Elements, Electrical resistance- Ohm's law. Construction of circuit model. Voltage and potential in electric circuit. Current source. Kirchoff's laws. Analysis of a Circuit Containing Dependent Sources. Complex DC circuits. DC circuit analysis using SPICE.
Course objectives: The purpose of the course is to introduce the basic principles of electrical field and DC current circuits.
Learning outcomes: On successful completion of the course, students will be able to:

1. Understand fundamental laws of electricity.
2. Apply the fundamental laws of electricity for solving electric field problems.
3. Apply MATLAB software package for solving basic problems in the electrical field.
4. Understand and apply methods for DC circuit analysis such as: Kirchhof's lows, node
voltage method, mesh current method, superposition method, Thevenin's and Norton's theorem.
5. Understand transient response of first order circuits (series RC circuits).
6. Apply PSPICE Software for DC circuit analysis.
7. Apply gained knowledge in other electrical engineering fields.

Teaching Methodology: Presentations, tutorials, discussions, and laboratory works meaning: 60 hours presentations + 15 hours of tutorials, and 15 hours laboratory works.
Evaluation: First assessment:30%, Second assessment: 25%, Home exercises 10%, Attendance: 5%, Final exam, 30%, Total:100%.
Concretization tools: Computer, video projector, and equipped lab with necessary devices to illustrate all teaching material.
Literature:
1. Nexhat Orana, Bazat e elektroteknikës 1, Prishtinë, 1994
2. M.N. Sadiku, *Elements of electromagnetics*, Oxford University Press, New York, Seventh Edition, 2018
3. C. Alexander, -*Electric Circuits*, McGraw Hill, New York, 2000

Course title: Fundamentals of Programming (Mandatory, Sem. I, 5 ECTS)
Lecturer: Prof.Ass.Dr. Avni Rexhepi, Prof.Asoc.Dr. Kadri Sylejmani

The goal: The purpose of the course is to introduce the basic principles of programming and algorithms, for solving problems with computers and writing of respective programs (in C++ programming language).

Learning outcomes: On successful completion of the course, students will be able to write programs in C++ programming language, including the use of different functions.

Course content: Introduction to algorithms: sum, product and factorial. Array operations and algorithms: sum/product of the array members, counting members, searching and sorting. Matrix manipulations: creation, sum/product of the members, searching, creating vector from the matrix members. Introduction to C++ programming language: variable types, operators, statements. Input and output: input of numbers, arrays, strings and their output, use of manipulators. Branching: if, go to, switch(). Loops: while, do-while, for. Loops: breaking, continuing and exiting. Functions: defining and executing, parameters and arguments, mathematical and string functions. Using vectors and matrices with functions. Recursion, pointers, references. User defined types: enumerations, structures, classes.

Methods of teaching: 30 hours of lectures + 30 hours of auditory/lab exercises. Approximately 65 hours of personal study and exercise including home problems/tasks.

Grading System: Attendance lectures 10%, Attendance lab. 10%, Mid-term problems 30 %, Colloquy/Final Exam 50 %

Literature:

1. Agni Dika, "Algoritmet, me programe në C++", Universiteti i Prishtinës, Fakulteti Elektroteknik, Prishtinë, 2004, http://www.agnidika.net/algoritmetCpp.pdf
2. Agni Dika Bazat e Programimit në C++, Universiteti i Europës Juglindore, Tetovë, ISBN 9989-866-23-6, http://www.agnidika.net/programimiCpp.pdf
3. H.M. Deitel, P. J. Deitel, How to Program C++, Prentice Hall, Upper Saddle River, New Jersey, ISBN 0-13-111881-1
4. Robert Lafore, Object-Oriented Programming in C++, Sams, Indianopolis, Indiana, ISBN-10:0-672-32308-7

Course title: Communication skills (Mandatory, Sem. I, 3 ECTS)

Lecturer: Prof. Dr. Blerim Rexha & Prof. Dr. Sabrije Osmanaj,

The goal: The goal of this course is to develop written and verbal communication skills and group work.

Learning outcomes: On successful completion of the course, students will be able to:

1. write different official and business letters;
2. write formal and informal emails,
3. using social media;
4. write a five-paragraph essay;
5. analyze different reports (visit r., field r., feasibility r., progress r.);
6. write laboratory reports;
7. use the Internet to find specific information;
8. use the computer to write different reports;
9. write minutes of meetings;
10. write a paper on a particular problem or issue;
11. write CVs and applications for work;
12. hold oral presentations;
13. respond to job interviews.

Course content: Introduction to communication skills. Words and sentences. Writing process. Technical information. Internet information and e-mail: World Wide Web (www) and social media. Essays and exam answers. Laboratory reports. Reports. Proposals, specifications and manuals. Final year projects and reports. Spoken presentations. Group work and meetings. Letters. CVs and job applications. Interviews: Interview for job.

Methods of teaching: 30 hours of lectures + 15 hours of exercises. Approximately 80 hours of personal study and exercise including homework.

Grading System: Class activity 10%, three homework assignments 40%, online tests or final exam 50%

Literature:

1. Mike Markel & Stuart A. Selber, Technical Communication, 12th Edition, MacMillan, 2018
2. John W. Davies, Communication Skills. A Guide for Engineering and Applied Science Students, Prentice Hall, 2011.

Course: Practical in Mathematics (Elective, Sem. I, 3 ECTS)
Lecturer: Prof. Asoc. Dr. Valdete Rexhbeaj-Hamiti

Short course description: In this course will be studied algebraic expressions, polynomials, equations with an unknown, inequations, arithmetic and geometrical strings, trigonometry and analytic geometry.

Course objectives: Students should be trained so that the knowledge gained through this course can be applied in acquiring knowledge from the following subjects.

Learning outcomes:
Upon completion of this course the student will be able:

1. to design and solve different problems in the field of equations with an unknown and their implementation;
2. to operate with polynomials;
3. to apply basic concepts from analytical geometry and trigonometry to various engineering problems.

Teaching methodology: Lectures, discussions, exercises, consultations, homework, seminar work, intermediate evaluations, final exam.
Evaluation methods: Seminar work and homework (20%), first intermediate evaluation (30%), second intermediate evaluation (30%), final exam (20%).
Concretization tools/IT: Pencil, whiteboard, projector and computer.
Ratio between theoretical and practical part: Ratio between the theoretical and practical part is 1:1.lul

Literature:

1. Hamiti E., Peci H., Loshaj Z., Gjonbalaj Q., Lohaj Sh. - Përmbledhje detyrash nga matematika, Prishtinë 2001.
2. M. Berisha, D. Kamberi, R. Gjergji, R. Zejnullahu, Përmbledhje detyrash nga matematika, Prishtinë 1990

Course: Basic Software Tools (Elective, Sem. I, 3 ECTS)
Lecturer: Core FECE staff

Short course description: This course provides a comprehensive introduction to fundamental software tools used across various professional domains. Covering operating systems, file management, text editors, version control, productivity tools, programming fundamentals, and data management, the course aims to equip students with practical skills for efficient work in diverse environments.
Course objectives: Familiarize students with a range of software tools, provide hands-on experience in file management, coding, collaboration, and develop proficiency in text editors, version control, productivity tools, programming fundamentals, and data management.
Learning outcomes: Understand roles of common software tools, develop practical skills in file management, coding, and collaboration, proficiently use text editors, version control, and productivity tools, gain basic understanding of programming and data management, effectively use collaboration tools for communication and file sharing.
Teaching methodology: 30 hours of lectures + 15 hours of exercises. Approximately 65 hours of personal study and exercise including homework
Evaluation methods: Class activity 10%, three homework assignments 40%, online tests or final exam 50%
Concretization tools/IT: Pencil, whiteboard, projector and computer.
Ratio between theoretical and practical part: Ratio between the theoretical and practical part is 1:1.

Literature:

- David Thomas, Andrew Hunt, The Pragmatic Programmer: Your Journey To Mastery, 20th Anniversary Edition (2nd Edition), Addison-Wesley Professional, 2019.
- George Beekman and Ben Beekman, Digital Planet: Tomorrow's Technology and You, Introductory (10th Edition) (Computers Are Your Future), Prentice Hall, 2011.
- Michael J. Quinn, Computer Confluence Complete: Tomorrow's Technology And You 7th Edition, Prentice Hall, 2005.

2. M.N. Sadiku, Elements of electromagnetics, Oxford University Press, New York, Seventh Edition, 2018
3. C Alexander, M.N. Sadiku, Electric Circuits, McGraw Hill, New York, 2000.

YEAR 1
SEMESTER 2

Course title: Analytic Geometry with Calculus 2 (Mandatory, Sem. II, 7 ECTS)
Lecturer: Prof. Asoc. Dr. Qefsere Gjonbalaj, Prof. Asoc. Dr. Shqipe Lohaj, Prof. Asoc. Dr. Valdete Rexhbeqaj-Hamiti

Course description: In this subject we work: Analytic geometry in space, integral computation and its implementation, functions with two or more variables and ordinary differential equations.
Course objectives: The student should be able to apply the knowledge gained through this course as an auxiliary device in the studies of electrical engineering and computer engineering subjects.
Expected results: After completion of the course, student will be able to:

1. understand the concept of indefinite and definite integral;
2. apply the concepts of integrals in the calculation of various measures in Geometry, and in various fields of engineering;
3. generalize concepts related to functions with one variable into multi variable functions and in particular into those with two variables;
4. know the concept of differential equations of the first order and of higher orders and know how to find their solutions;
5. apply differential equations in solving various practical problems.

Methodology of teaching: Lectures, discussions, exercises, consultations, homework, mid term exams, final exams.
Methods of assessment: Attendance and classroom activity (10 %), Homework and seminar (10%), First periodic exams (25%), Second periodic exams (25%), Final exams (30%).
Concretization tools: pencil, whiteboard, projector and compjuter.
Ration between Theoretical part and exercises: 2:1
References

1. Hamiti E. - Matematika I, Prishtinë 2008.
2. Hamiti E. - Matematika II, Prishtinë 2008.
3. Hamiti E. - Matematika III, Prishtinë 2008.
4. Peci H, Doko M. - Përmbledhje detyrash të zgjidhura nga Matematika I, Prishtinë 1997.
5. Loshaj Z. - Përmbledhje detyrash të zgjidhura nga Matematika II, Prishtinë 1996.
6. Hamiti E, Lohaj SH.- Përmbledhje detyrash të zgjidhura nga Matematika III, Prishtinë 2008.

Course title: Physics for Engineering 2 (Mandatory, Sem. II, 5 ECTS)
Lecturer: Dr. Sc. Valon Veliu

Short course description: The course includes basic knowledge of physics necessary to gain general knowledge that is basic in engineering.
Course objectives: Using the physics laws of modern physics in modelling and solving specific engineering problems.
Learning outcomes: On successful completion of the course, students will be able to:

- analyze and solve problems from the elasticity of solid bodies;
- apply the linearization technique to equations of motion of oscillatory;
- explain the wave equation in different mediums;
- understand the equation of mechanical waves and the whole problem for different environments;
- analyze and solve problems from acoustics;
- analyze and solve problems for geometric and wave optics, reflection and refraction of light Snell's law, interference and diffraction;
- analyze and solve problems for the quantum nature of light, black body radiation, photoelectric effect;
- analyze and solve problems for the structure of the atom and atomic models;
- analyze and solve problems from radioactivity and special relativity.

Teaching methodology: Lectures, numerical exercises, laboratory exercises, seminar, homework, consultations.

Evaluation methods: Classroom activity 5%, Laboratory exercises 7,5%, Homework 7,5%, Seminar work 10%, First periodic evaluation 35 %, Second periodical evaluation 35% or Final Exam 70 %.
Concretization tools/IT: Pencil, whiteboard, projector and computer.

Literature:

1. S. Skenderi, R. Maliqi, "Fizika për studentët e fakulteteve teknike", UP, Prishtinë, 2005..
2. David Halliday, Robert Resnick, Jearl Walker, Fundamentals of Physics Extended, 10th edition, Wiley (2013).
3. Raymond Serway and John Jewett, Physics for Scientists and Engineers with modern physics, 10th edition, Cengage Learning, (2018).
4. James S Walker, - Physics, 5th edition, Pearson Addison-Wesley (2016).

Course title: Fundamentals of electrical engineering 2 (Mandatory, Sem. II, 6 ECTS)
Lecturer: Prof. Dr. Enver Hamiti, Prof. Dr. Mimoza Ibrani, Prof. Ass. Dr. Vjosa Shatri

Short Course description: Basics of magnetism. Magnetic flux density. Lorentz force. Biot-Savart's law. Forces due to magnetic field. Magnetic torque and moment. Ampere's law. Magnetic dipole. Magnetization in materials. Generalized Ampere's law. Magnetic boundary conditions. Magnetostatic field analysis using MATLAB. Faraday's law. Inductors and inductances. Magnetic energy. Magnetic circuits. Current and voltage waveforms. Techniques of Circuit Analysis. Sinusoidal Steady-State Analysis. The Sinusoidal Response, The Passive Circuit Elements in the Frequency Domain, Sinusoidal Steady-State Power Calculations, Maximum Power Transfer. AC circuit analysis using SPICE. Magnetically coupled circuits. Response of First-Order *RL* and *RC* Circuits. Transient circuit analysis using SPICE. Balanced Three-Phase Circuits. SPICE analysis of three phase circuits.
Course objectives: The purpose of the course is to introduce the basic principles of magnetic field and AC current circuits analysis.
Learning outcomes: On successful completion of the course, students will be able to:

1. Understand fundamental laws of magnetism.
2. Apply the fundamental laws of magnetism for solving of magnetic field problems.
3. Apply MATLAB software package for solving basic problems in magnetic field.
4. Understand and apply methods for AC circuit analysis such as: Kirchhof's lows, node voltage method, mesh current method, superposition method, Thevenin's and Norton's theorem.
5. Understand transient response of first order circuits (series RL circuits).
6. Apply PSPICE Software for AC circuit analysis.
7. Apply gained knowledge in other electrical engineering fields.

Teaching Methodology: Presentations, tutorials, discussions, and laboratory works meaning**:** 60 hours of lectures + 15 hours of tutorials, and 15 hours laboratory works.
Evaluation: First assesment:30%, Second assesment: 25%, Home exercises 10%, Attendance: 5%, Final exam, 30%, Total:100%.
Concretization tools: Computer, video projector, and equiped lab with neccessary devices to ilustrate all teaching material.
Literature:

1. Nexhat Orana, Bazat e elektroteknikës 2, Prishtinë, 1994
2. M.N. Sadiku, Elements of electromagnetics, Oxford University Press, New York, Seventh Edition, 2018
3. Ch. Alexander, M.N. Sadiku, Electric Circuits, McGraw Hill, New York, 2000.

Course title: Algorithms and Data Structures (Mandatory, Sem. II, 6 ECTS)
Lecturer: Prof.Asoc.Dr. Kadri Sylejmani, Prof.Ass.Dr. Avni Rexhepi

The goal: The purpose of the course is to help students in advancing their knowledge for different algorithms, data structures and use of classes and objects in programming.

Learning outcomes: On successful completion of the course, students will be able to compile advanced algorithms, to define structures and classes and to use them in writing advanced programs.

Course content: Definition and use of advanced functions, inline functions, macro functions, function overloading, templates. User defined types. Object oriented programming. Structures: definition of different structures, with different functions as their components, operating structure components. Classes and objects: class definition, classes and member functions. Using public and private members. Declaring objects and operating with their components. Class constructor and destructor. Inheritance, arrays within objects and arrays of objects. Pointers and functions with pointers. References and functions with references. Algorithms, analysis of algorithms, algorithm's growth rate, classification. Data structures. Stack. Queue. Linked lists, adding/deleting nodes. List searching and sorting. Binary tree, tree traversal algorithms, insertion, search and deletion, BST-binary search tree, heap, balanced trees. Graphs, traversal algorithms, minimum spanning tree (Dijkstra-Prim, Kruskal), shortest path algorithm (Dijkstra). Searching and sorting algorithms (Insertion sort, Selection sort, Bubble sort, Shell sort, Merge sort, Quick sort, Heap sort, Bucket sort, Radix sort).

Methods of teaching: 30 hours of lectures + 30 hours of auditorial/lab exercises. Approximately 65 hours of personal study and exercise including home problems/tasks.
Grading System: Attendance lectures 10%, Attendance lab. 10%, Mid-term problems 30 %, Collocui/Final Exam 50 %
Literature:

1. Agni Dika, Programimi i Orientuar në Objekte, me programe në C++, UEJL, Fakulteti i Shkencave Bashkëkohore, Tetovë, ISBN 9989-866-25-2, http://www.agnidika.net/programimiobjekte.pdf
2. D. S. Malik, C++ Programming: Program Design Including, Data Structures, Course Technology, Thomson Leraning, Boston, Massachusetts, ISBN 0-619-03569-2
3. H.M. Deitel, P. J. Deitel, How to Program C++, Prentice Hall, Upper Saddle River, New Jersey, ISBN 0-13-111881-1
4. Robert Lafore, Object-Oriented Programming in C++, Sams, Indianopolis, Indiana, ISBN-10:0-672-32308-7
5. D. S. Malik, Programming: From Problem Analysis To Program Design, Course Technology, Thomson Leraning, Boston, Massachusetts, ISBN 0-619-06213-4

Course title: Digital logic circuits (Mandatory, Sem. II, 6 ECTS)
Lecturer: Prof. Dr. Sabrije Osmanaj, prof. Ass. Dr. Artan Mazrekaj

Short course description: Number systems and codes: number systems, codes, encoding, error detection and correction. **Boolean algebra:** propositional logic, Boolean variables, basic operations, derived operations, axioms and theorems, proofs of theorems. **Boolean functions and logic gates:** representations, methods of simplification and conversion, Karnaugh map and truth table, logic gates and circuits, functionally complete sets of operations, timing hazards, logic families and technologies and their characteristics.
Combinational logic circuits: encoders and decoders, multiplexers and demultiplexers, comparators, adders, multipliers, arithmetic and- logic units. **Computer-aided digital design:** minimizers, schematic editors, circuit simulators, hardware description languages, PCB layout designers, IC layout designers. **Sequential logic circuits:** latches and flip-flops, truth table and excitation table, registers, counters, shift registers, ring counters, excitation equations, state table and state diagram, analysis and synthesis of sequential logic circuits. **Three-state buffers and buses:** buffer, Schmitt trigger buffers, three-state buffers, serial buses, parallel buses. **Programmable logic circuits:** storage matrix, ROM, PROM, EPROM, EEPROM, Flash, PLA, PAL, GAL, SRAM, DRAM, CPLD, FPGA. Use of hardware description languages for implementation of combinational and sequential logic in CPLD and
FPGA circuits **Additional topics (in case of spare time, but not required in exams):** microcontrollers, microprocessors, analog-digital and digital-analog converters, clock generators.
Course objectives: To gain the basic theoretical understanding of functioning of digital structures. To acquire the knowledge and basic experience of practical design, implementation and testing of digital structures.
Learning outcomes: After successful completion of the course, students will be able to: describe the basic combinational and sequential structures of digital circuits; explain the functioning of such structures; describe this functioning in one of the hardware description languages; detect flaws in such functioning; eliminate the basic among such flaws; choose the optimal design of a digital structure, taking into account the requirements of size, cost and reliability of functioning; design combinational and sequential circuits using the methods of abstract and structural synthesis; build a prototype of the designed structure in the form of a printed circuit board; test the functioning of such a prototype; Formulate the strengths and weaknesses of the chosen design.
Teaching methodology: Lecture classes with examples of problem solving and tutorials to illustrate the theoretical concepts, laboratory work for acquisition of practical skills in design, implementation and testing of digital structures.
Evaluation methods: Class attendance 10%; Assessment from tests 90%. Total: 100%
Concretization tools/IT: Computer, projector, lab, table.
Ratio between theoretical and practical part: Ratio between the theoretical and practical part is 1:1.
Literature:

1. Floyd Thomas L., Digital Fundamentals (10th Edition), Prentice Hall, 2008.

2. M. Morris Mano, M. D. Ciletti. Digital Design, 6th ed. Pearson/Prentice Hall, 2017.
3. Fundamentals of Digital Circuits, 3rd Edition, by A. ANAND KUMAR, 2014, Delhi.
4. S.M. Deokar, A. A. Phadke, "Digital Logic Design and VHDL", Wiles, 2009
5. Digital Circuit Analysis and Design with SIMULINK Modeling: And Introduction to CPLDs and FPGAs, Second Edition, Steven T. Karris, Orchard Publications 2007.
6. J. F. Wakerly. Digital Design: Principles and Practices, 5th ed. Pearson/Prentice Hall, 2017.
7. C. Maxfield. Bebop to the Boolean Boogie, 3rd ed. Newnes, 2009.

YEAR 2

SEMESTER 3

Course: Discrete Mathematics and Probability (Mandatory, Sem. III, 5 ECTS)
Lecturer: Prof. Asoc. Dr. Qefsere Doko Gjonbalaj

Course status: Mandatory
Course Description: In this course parts from Mathematical Logic, sets, some important functions, and algebraic structures will be studied. Basic concepts of numerical series, polynomial series and functional series. Fourier series. Special attention is paid to the part of the Graph Theory (graphs and trees) and Discrete Probability, where the basic concepts of probability, discrete and continuous random variables will be studied. Numeric parameters of random variables, variance, standard deviation. Then will be explained the discrete distributions, distribution functions and covariance. Basic elements of mathematical statistics, the role of statistics in engineering and statistical tests.

Course objective

Students should be trained so that the knowledge gained through this course can be applied as auxiliary equipment in electrical and computer engineering studies.

Learning outcomes:
Upon completion of this course the student will be able to:

1. Understand the role of mathematical logic, finite sets and relations
2. Become familiar with the problems associated with n-ary relations
3. Analyze electrical engineering problems using Furie series
4. Apply Fourier series into concrete professional problems
5. Solve some characteristic problems in graph theory
6. Make mathematical models related to concrete professional problems, using knowledge from graph theory and trees
7. Know the basic concepts of the Probability Theory and Statistics;
8. Know the basic distributions attributes as well as their application to professional problems;
9. Process and present statistical data

Teaching methodology: Lectures, discussions, exercises, consultations, homework, seminar work, intermediate evaluations, final exam.
Evaluation methods: Seminar work and homework (10%), first intermediate evaluation (25%), second intermediate evaluation (25%), final exam (40%).
Concretization tools/IT: Pencil, whiteboard, projector and computer.

Ratio between theoretical and practical part: Ratio between the theoretical and practical part is 2:1.

References

1. Qefsere Doko Gjonbalaj, *"Matematika III me Matematikë Diskrete (Programi i Kompjuterikës)"*; Libër universitar; Universiteti i Prishtinës-FECE; 2017, Prishtinë; ISBN 978-9951-00-132-8; Botimi i dyte
2. Qefsere Doko Gjonbalaj , Shukri Haxha, *" Permbledhje Detyrash nga Matematika III me Matematikë Diskrete (Programi i Kompjute rikës)"*; Libër universitar; Universiteti i Prishtinës-FECE; 2017, Prishtinë; ISBN 978-9951-00-132-8
3. Kenneth H. Rosen; MC GRAW HILL: Discrete Mathematics and its Applications, Fifth Edition 2003, ISBN 0-07-242434-6; USA
4. Kenneth H. Rosen; MC GRAW HILL: Student Solutions Guide for Discrete Mathematics and its Applications, Fourth Edition 2003, ISBN 0-07-289906-9; USA
5. T.T. Soong "Fundamentals of Probability and Statistics for Engineers" State University of New York at Buffalo, Buffalo, New York, USA, 2004

Course title: Electronics (Mandatory, Sem. III, 5 ECTS)
Lecturer: Prof. Dr. Qamil Kabashi, Prof. Dr. Milaim Zabeli
Course goal: The aim of the Electronics course is to furnish students with a thorough comprehension of fundamental electronic components and circuits, specifically tailored to their application in computer systems. Encompassing semiconductors, diodes, rectifiers, wave shaping networks, field-effect and bipolar transistors, the course is designed to empower students with essential knowledge for the analysis and design of electronic circuits within the context of computer systems.

Learning outcomes: On successful completion of the course, students will be able to:

- Comprehend the function of diodes, rectifiers, and wave shaping circuitss, and identify their practical applications in computer engineering, particularly in signal processing technology.
- Develop knowledge of the application and function of field-effect transistors (MOSFET) and bipolar transistors in the context of computer systems.
- Understand the role and utilization of transistors as switches in computer engineering applications, including a thorough analysis within the scope of the course.
- Apply small signal AC analysis to single-stage amplifiers using BJT and MOSFET.
- Understand the structure and function of memory circuits such as bistable circuits, within the context of computer architecture.
- Understand various types and architectures of semiconductor memories, with a focus on SRAM cells, DRAM cells, MOS ROM and flash memory in the context of computer systems.
- Understand the function and applications of operational amplifiers in computer engineering, including practical analysis scenarios relevant to the field.

Course content: Semiconductors. Diodes, rectifiers, and wave shaping networks. and Bipolar and Field effect transistors. The Transistor as swich. Small signal AC analysis of single-stage amplifier with BJT/MOSFET Memory circuits (bistable circuits). Semiconductor Memories- Types and Architectures- SRAM and DRAM cells. The Row-Address Decoder and Column address decoder, Pulse-Generation Circuit, A MOS ROM). Operational amplifiers and applications.

Evaluation method: Test 1: 20%, Test 2: 20%, Laboratory exams 30%, Final exam 20%, Attendance to lectures 10%. Success in preliminary assessments is a prerequisite for the final exam.

Literature:

1. Donald Neamen, Microelectronics: Circuits Analysis and Design, McGraw-Hill Education, 4th Edition, 2010.
2. Adel S. Sedra, Kenneth C. Smith, *Microelectronic Circuits*, 8th edition. Oxford Univ. Press, 2019.
3. Myzafere Limani, Qamil Kabashi, Elektronika (pjesa e pare), Universiteti i Prishtinës, 2023.

Course title: Legal, Ethical and Social Issues in ICT (Elective, Sem. IV, 4 ECTS)

Lecturer: Prof. Ass. Dr. Dhuratë Hyseni
Course goal: The purpose of the course is to motivate and apply the concept of critical thinking and information technology's impact on personal, societal, legal and ethical issues.
Learning outcomes: On successful completion of the course, students will be able to:

1. Think critically about information technology issues, actively engage others in dialogue about them, and relate them to personal and societal values.
2. Analyzing and critically evaluate ideas, arguments, and points of view.
3. Express a reasoned position on an issue, both orally and in writing.
4. Analyze the relationships among ethical, social, and political issues that are raised by information systems.
5. Identify the main moral dimensions of an information society and specific principles for conduct that can be used to guide ethical decisions.

Course content: Introduction to Ethical Theory; Professional Ethics and Responsibility; Critical Thinking Skills; Technology and Privacy; Technology and Free Speech; Encryption and Communication; Intellectual Property; Cybernetic Crime, Security and Protection; Moral Dimension of Information System; Social Issues and Technology; Professional Ethics.
Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 65 hours of personal study and exercise including seminars.
Grading System: Seminar 10%, Mid-term exams 30 %, Final Exam 60 %
Literature:

1. Sara Baase, A Gift of Fire: Social, Legal, and Ethical Issues for Computing Technology, 4/E, ISBN-10: 0132492679
2. Joseph Migga Kizza, Ethical and Social Issues in the Information Age, fourth edition, ISBN 978-1-84996-037-3
3. M. David Ermann, Michele S. Shauf; Computers, Ethics, and Society latest Edition, Oxford University Press, ISBN: 0195143027

Course title: Databases (Mandatory, Sem. III, 5 ECTS)
Lecturer: Prof. dr. Lule Ahmedi
The goal: Students will be able to apply the basic concepts and techniques for the design and implementation of database applications.
Learning outcomes: Upon completion of this course, the student shall be able to:

1. Know the techniques of describing and managing a huge set of data based on the relational model.
2. Use the experience gained here to design and implement database projects in practice.
3. Know the typical problems of databases in the practice and compare existing systems in order to make the proper solution to overcome those problems.
4. Identify the different relevant aspects during the development of huge projects of databases, and respectively organize the adoption of basic principles.

Course content: A preliminary list of topics cover (mainly based on the audience): Introduction and relational model. Relational algebra. Datalog: Logical rules. SQL: Simple queries, aggregation, grouping, set operators, embedded queries, database creation and views, population with data. SQL and programming languages. SQL: Integrity and triggers. SQL: Procedures and functions. SQL: Data security. Conceptual design: ER model and transformation into relational model. Formal design: functional dependencies and normal forms. Physical design. Operator / query evaluation. Transaction management. Crash recovery.
Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 65 hours of personal study and exercise including project work.
Grading System:
Active attendance 10%, Project assignment 3 x 10% = 30%, Final exam 60 %.
Literature:

1. Database Management Systems (3rd Edition). Raghu Ramakrishnan, Johannes Gehrke. McGraw Hill, 2002.
2. Database Systems: The Complete Book (2nd Edition). Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom. Pearson, 2008.
3. Database System Concepts (7th Edition). Avi Silberschatz, Henry F. Korth, S. Sudarshan. McGraw-Hill, 2019.

Additional literature:

4. A guide to the SQL standard. C. J. Date, Hugh Darwen. Addison-Wesley Professional, 1996.

SQL for Web Nerds. Philip Greenspun. http://philip.greenspun.com/sql/

Course title: Object Oriented Programming (Mandatory, Sem. III, 5 ECTS)
Lecturer: Prof. Dr. Isak Shabani
Course goal: This course enables students to be prepared and successfully apply the concepts and techniques of programming with objects, enabling students to apply object-oriented techniques in software projects. The purpose of the course is to prepare students with modern knowledge in "thinking in object programming", a precondition necessary for complex software systems based on object-oriented programming.

Learning outcomes: On successful completion of the course, students will be able to:

1. Understand main concepts of object-oriented programming;
2. To write code with classes and use objects;
3. To use inheritance, polymorphism, encapsulation and abstraction;
4. To be able to predict exceptions and error handling;
5. To be able to create abstract classes and interfaces;
6. To be development applications with graphical user interface;
7. Realize e project relating a particular issue using object-oriented programming.

Course content: Basic principles of object-oriented programming and design. Modelling. Reusability. Frameworks for program development. Introduction to Java. Classes and objects. Java packages. Constructors and destructors. Reference and value types, Data access. Attributes, methods and operators. Delegates and events. Encapsulation, class inheritance, polymorphism. Abstract classes. Interfaces. Introduction to design patterns. Exceptions and error handling. Generic types and methods. Class collection. Multithreading and multithreaded applications. Development of applications with graphical user interface. Events. Simple and complex graphical components. Development of custom graphical components. Program testing. Program performance analysis. Program optimization.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 65 hours of personal study and exercise including homework.

Grading System:
Attendance 10%, Practical project 30 %, Final Exam 60 %

Literature:

1. C. Thomas Wu, "An Introduction to Object-Oriented Programming with Java", 2009, Publisher: McGraw-Hill Education; 5th edition
2. Daniel Liang, “Introduction to Java Programming”, 2014, 10th Edition, Armstrong Atlantic State University
3. Danny Poo, "Object-Oriented Programming and Java", 2007, Second Edition 2nd Edition
4. Richard Naoufal, "Java: Object-Oriented Programming Concepts", 2018.

Course title: Client-Side Web Programming (Mandatory, Sem. III, 5 ECTS)
Lecturer: Prof. Asst. Dr. Dhurate Hyseni

The goal: Students will be able to understand basic concepts related to client-server programming paradigms actual on the Web, as well as to provide experience in design and implementation of Web-based systems.

Learning outcomes: Upon completion of this course, the student shall be able to:

1. Demonstrate fundamental knowledge on traditional as well as cutting-edge concepts and technologies on the World Wide Web.
2. Develop Web applications in practice – prior to that, choose the architecture, modeling and techniques that fit to the nature of the application to develop.
3. Distinguish among the advantages and drawbacks of client versus server paradigms of distributing software components in the Web.

Course content: A preliminary list of topics cover (mainly based on the audience): Introduction to the Internet and the WWW. Client-side programming: HTML, CSS, DOM model, and JavaScript. Interfaces: jQuery. Reuse of existing API-s. Added functionality in Web: wikis, blogs/RSS, tagging. Server-side programming: PHP. Databases on the Web: PHP and MySQL. Regular expressions. Cookies. Session control. Re-engineering an application on the Web.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 65 hours of personal study and exercise including project work.

Grading System:

Active attendance 10%, Project assignment 4 x 10% = 40%, Final Exam 50 %.

Literature:

1. Fundamentals of Web Development (2nd Edition). Randy Connolly, Ricardo Hoar. Pearson, 2018.
2. PHP and MySQL Web Development (5th Edition). Luke Welling, Laura Thompson. Addison-Wesley Professional, 2016.
3. Programming the World Wide Web (8th Edition). Robert Sebesta. Addison Wesley, 2014.
4. TCP/IP Illustrated, Volume 1: The Protocols (2nd Edition). Chapter 1: Introduction. Kevin R. Fall, W. Richard Stevens. Addison-Wesley Professional Computing Series, 2011.
5. Unleashing Web 2.0: From Concepts to Creativity. Gottfried Vossen, Stephan Hagemann. Morgan Kaufmann, 2007.
6. A number of resources on the Web.

YEAR 2

SEMESTER 4

Course title: Computer Architecture (Mandatory, Sem. IV, 6 ECTS)
Lecturer: Prof. Dr. Qamil Kabashi

The goal: The objective of this course is to provide students with a fundamental understanding of the functional components of a computer system. The focus of the course is on the hardware aspects of a system during the execution of software.

Course content: Introduction to computer organization and architecture. Designing for Performance. Pentium and PowerPC evolution. Top-level view of computer function and interconnection structures. BUS interconnection. Cache memory and elements of cache design. Internal memory. Input/output devices. Programmed and interrupt driven I/O. Direct memory access. RISC versus CISC architecture. CPU structure and function, Instruction Pipelining, Multi-core processors. Parallel processing and multiple processor organization.

Learning outcomes: On successful completion of this course, students should be able to:

- Explain computer architecture, organization, and structure relating to contemporary design for performance.
- Analyze the program execution process through instruction cycles, and the functioning of hardware components during this process using von Neumann architecture as a model.
- Classify various types of buses, their signals, and the different arbitration techniques.
- Illustrate internal and cache memory mapping including parameters of cache design.
- Differentiate and analyze interrupt-driven I/O, programmed I/O, and direct memory access mechanisms
- Explain the different architectural and organizational design issues that can affect the performance of a computer such as Pipelining, Superscalar architecture, RISC, Multi-core processors, and Multiple Processor Organization.
- Implement a simple assembly program and C/C++ using development plates (ARM, ARM-Coretex, 8051).
- Complete assignments, experiments in the laboratory, and present a technical report.

Teaching methodology: Lectures, auditoria exercises, Assignments, Lab Experiments, Lab report and presentation.

Assessment: First midterm exam 25%, Second midterm exam 25 %, Lab experiments (20%), Seminar paper 20%, Attendance 10%.

Concretization means/IT: Laptop, SMARTboard, and practical part will be done in the laboratory

Ratio between the theoretical and practical part of teaching: 40:60

Literature:

1. W. Stallings, Computer Organisation and Architecture, 11th Edition, Pearson, 2021
2. S. Tanenbaum, Structured Computer Organization, 6th Edition, Pearson, 2013

Course title: **Computer Networks** (Mandatory, sem IV, 5 ECTS)
Lecturer: Prof. Dr. Blerim Rexha
The aim of the course (module): Introduction to ISO model layers in computer networks, familiarity with communication protocols, description and operation of services at the application, transport, network, and physical layers.

Learning outcomes: After completing this course (module) the student will be able to:

1. Understand principles of protocol layers and services,
2. Understand and analyze TCP / IP protocols,
3. Apply TCP / IP protocols,
4. Analyze and design network configuration,
5. Apply basic principles of distributed applications in networks,
6. Evaluate and design applications based on TCP / IP protocols.

Course content: layer protocols and services, Internet Service Provider (ISP), Internet History. Principles of network applications, Web, HTTP, FTP, email, DNS and web server. Introduction to the transport layer services, multiplexing and de-multiplexing, reliable data transfer, TCP Protocol, wireshark tool for monitoring traffic, routing, network service model, virtual circuits and datagram networks, ports, IP protocol, routing algorithms, routing in the internet, link layer services, error detection codes, CRC, MAC, LAN, Ethernet Hubs & Switches, PPTP protocol, CDMA, WiFi, Mobile IP, cellular architecture, small offices home office network configuration (SOHO)

Methods of teaching: 30 hours for lectures, 30 hours for laboratory exercises. Approximately 65 hours of personal study including home/group project assignments.

Grading System: Classroom Assessment 10%, Projects 40%, Final assessment 50 %

Literature:

1. James F. Kurose & Keith W. Ross, "Computer Networking", 7th Ed., Pearson Inc., 2017
2. Dougles Comer, "Internetworking with TCP/IP, Principles, Protocols, and Architecture", 2013

Course title: Data Security (Mandatory, Semester IV 5 ECTS)

Lecturer: Prof. Dr. Blerim Rexha

The goal: To provide students with practical survey of principles and practice of cryptography and data security, smart cards and their practical use.

Learning outcomes: On completion of this course, students should be able to:

1. Understand basic principles about cryptography,
2. Understand basic principles about symmetric and asymmetric encryption algorithms,
3. Explain basic principles of hash algorithms,
4. Apply different cryptographic and hash algorithms,
5. Understand about smartcards and apply their usage in real life applications,
6. Evaluate and support public key schema, and
7. Analyze and design applications that use cryptographic algorithms.

Course content: Will include: Symmetric encryption algorithm: Data Encryption Standard (DES) and its variants, Asymmetric encryption algorithm: Rivest Shamir Addelman (RSA), Hash functions: MD5, SHA1 and SHA256, Public Key Infrastructure (PKI), Digital signatures, Authentication, Smart cards, Biometric documents, Hands on: using digital certificates for data security.

Methods of teaching: 30 hours of lectures + 30 hours of lab exercises. Approximately 65 hours of personal study including home/group project assignments.

Grading System: Classroom assessment 10%, Projects 40%, Final assessment 50 %

Literature:

1. Bruce Schneier, Applied Cryptography, ISBN 978-1-119-09672-6, March 2015
2. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanston, Handbook of Applied Cryptography, ISBN: 0-8493-8523-7 1996
3. H.X. Mel & Doris Baker, Cryptography Decrypted, 2004
4. Mtthew MacDonald & Erik Johansson: C# Data Security, 2003

Course title: Human Computer Interaction (Mandatory, Sem. IV, 5 ECTS)
Lecturer: Prof. Dr. Isak Shabani

Course goal: The aim of the course is to prepare students and enable them to understand the human-computer interactions concepts and techniques and equip students with modern knowledge in thinking of human connectivity with the computer through graphical interfaces and interaction design.

Learning outcomes: On successful completion of the course, students will be able to:

1. Understand main concepts of human-computer interactions;
2. To know and identify main characteristics of virtual reality;
3. To understand basic concepts on interactions design;
4. To understand human-computer communication paradigms;
5. To analyze, design, implement and evaluate software systems through techniques waterfall model, scrum and agile methods.

Course content: Introduction to the human-computer communication subject. Human. Thinking. Computer. Virtual reality. Human-computer interaction. Norman Stages. Paradigms in HCI. Interaction design. HCI in software processes. Basics of designing interactive systems. The design rules of interactive systems. Support and implementation of interactive systems. Interactive Systems Evaluation Techniques. The universal design of interactive systems. Supporting users in interactive systems.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 65 hours of personal study and exercise including homework.

Grading System:
Attendance 10%, Practical project 30 %, Final Exam 60 %

Literature:

1. 1. Alan Dix, Janet Finlay, Gregory D. Abowd and Russell Beale, "Human-Computer Interaction", 2004
2. Jenny Preece, Helen Sharp, Yvonne Rogers, "Interaction Design: Beyond Human-Computer Interaction", 2015
3. Donald A. Norman, "The Design of Everyday Things", Paperback 2013

Course title: Server-Side Web Programming (Mandatory, Sem. IV, 5 ECTS)
Lecturer: Prof. Asst. Dr. Dhurate Hyseni
The goal: This course covers advanced Web programming at the client and the server side, advanced and recent models, concepts, and techniques of development in the Web, as well as adaption to and from other platforms on the Web such as mobile devices and social networks.
Learning outcomes: Upon completion of this course, the student shall be able to: 1.

1. Demonstrate knowledge on advanced and latest concepts and techniques on the World Wide Web.
2. Develop advanced Web applications in practice.
3. Put in use advanced approaches of Web applications programming or configuration, such as through frameworks and design patterns.
4. Adapt Web solutions to and from other physical or conceptual platforms, like mobile devices or social networks.

Course content: Object-orientation on the Web. Asynchronous client-server communication: AJAX. Client-side frameworks: Bootstrap. Code re-use using design patterns: MVC. Configuration or programming: CMS-s (WordPress). Server-side frameworks: Laravel. Responsive programming and migration to mobile devices: CSS3 and new mobile behavior. Advanced multimedia (HTML5 canvas). Web security. Migrating a Web application to a social network. Integrate social networks to a Web application. Latest trends: Node.js and MongoDB.
Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 65 hours of personal study and exercise including project work.
Grading System:
Active attendance 10%, Project assignment 4 x 10% = 40%, Final Exam 50 %.
Literature:

1. Head First Ajax. Rebeca Riordan, O'Reilly Media, 2009.
2. Fundamentals of Web Development (2nd Edition). Randy Connolly, Ricardo Hoar. Pearson, 2018.
3. Programming the Mobile Web (2nd Edition). Maximiliano Firtman. O'Reilly Media, 2013.
4. Programming Social Applications. Jonathan LeBlanc, O'Reilly, 2011.
5. A number of resources on the Web.

Course title: Marketing for Engineers (Elective, Sem. 4, 4 ECTS)
Lecturer: Prof. Asst. Dr. Nora Sadiku
Course goal: The primary goal of this course is to equip engineering students with a nuanced understanding of marketing principles, emphasizing their application within technology-driven sectors. Students will gain insights into crafting effective marketing strategies for engineering products and services.
Learning outcomes: On successful completion of the course, students will be able to:

1. Define marketing and marketing process
2. Analyze markets and buyer behavior
3. Explain market segmentation, targeting and positioning
4. Understand the marketing mix including product, pricing, marketing channels and communication channels
5. Explain the importance of creating a competitive advantage, the global market as well as social responsibility and ethics in marketing

Course content: Defining marketing and marketing process, Marketing Environment, Managing marketing information, Markets and buyer behavior, Creating value for target customers, Products, Services, and Brands: Building Customer Value, Developing New Products and the Product Life Cycle, Pricing: Understanding and Capturing Customer Value, Marketing Channels: Delivering Customer Value, Engaging Consumers and Communicating Customer Value, Advertising and Public Relations, Personal Selling and Sales Promotion, Direct, Online, Social Media, and Mobile Marketing, Global marketplace. Social responsibility and Ethics

Methods of teaching: 30 hours of lectures + 15 hours of laboratory exercises. Approximately 55 hours of personal study and exercise including seminars.
Grading System: Seminar 10%, Mid-term exams 40 %, Final Exam 50 %
Literature:

1. Philip Kotler and Gary Armstrong. Marketing. UET Press. Tiranë. 2013
2. Philip Kotler and Gary Armstrong. Principles of Marketing. Ed. 18.. Pearson, 2021
3. Foundations of marketing William M.Pride, and O.C. Ferrell. (8th Ed.). Cengage learning, (2018).

Course title: Project Management (Elective, Sem. 4, 4 ECTS)
Lecturer: Prof. Asst. Dr. Nora Sadiku
Course goal: The main goal of this course is to empower students with the knowledge and skills needed to proficiently manage projects throughout their life cycle, addressing the specific challenges within their field.
Learning outcomes: On successful completion of the course, students will be able to:

1. Define key project management terms and concepts.
2. Recall the phases of the project life cycle and their significance.
3. Describe the project initiation process and its role in effective project execution.
4. Create a comprehensive project plan, including schedule, resource, and budget planning.
5. Assess the effectiveness of project closing and control mechanisms.

Course content: Introduction to project management, The Project Life Cycle (Phases), Project Initiation Process, Project planning, Planning and Defining the Project Scope, Project Schedule Planning, Resource Planning, Budget Planning, Procurement Management, Quality Planning, Risk Management Planning, Project execution, Project control, Project closing, Project presentations,

Methods of teaching: 30 hours of lectures + 15 hours of laboratory exercises. Approximately 55 hours of personal study and exercise including seminars.

Grading System: Seminar 10%, Mid-term exams 40 %, Final Exam 50 %

Literature:

1. Davies A. Igberaese . Introduction to Project Management A Source Book for Traditional PM Basics . Routledge 2023
2. A Guide to Project Management Body of Knowledge (PMBOK Guide) – and the Standard for Project Management [7 ed.2021]
3. Adriene Watt Project Management. Victoria, B.C.: BCcampus. 2014.

YEAR 3

SEMESTER 5

Course title: Operating systems (Mandatory, Sem. V, 6 ECTS)

Lecturer: Prof. Asst. Dr. Artan Mazrekaj

Course goal: The purpose of this course is to prepare students with modern knowledge of operating systems and enables to apply successfully the concepts of operating systems.

Learning outcomes: On successful completion of the course, students will be able to:

1. To understand main concepts and the structure of operating system;
2. To compare and make distinction among different operating systems;
3. To manage with process conflicts and executing threads;
4. To manage with memory, processor and input/output units;
5. To analyze operating systems and implement paperwork relating particular issues with operating systems.

Course content: Introduction to operating systems. Operating system structure, Process management: process concepts, threads, process scheduling, process synchronization, deadlocks. Memory management: memory management strategy, virtual memory management. I/O Management: file system, file system implementation, structure of disk data saving into. I/O systems, Operating System Protection. Operating System Security. Analysis and Studies on operating systems: UNIX, Linux, Windows, Minix, Mach, Android.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 65 hours of personal study and exercise including homework.

Grading System:
Attendance 10%, Seminar paper 10%, Project 20 %, Final Exam 60 %

Literature:

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, "Operating System Concepts", 10th Edition, 2018.
2. Andrew S. Tanenbaum, "Modern Operating Systems", 4th Edition, 2015.

Course title: Software Engineering (Mandatory, Sem. V, 5 ECTS)

Lecturer: Prof. Dr. Blerim Rexha

Course goal: The purpose of the course is to introduce the basic principles of the development of a cost-effective and high-quality software programs.

Learning outcomes: On successful completion of the course, students will be able to:

1. Understand basic principles about software engineering processes,
2. Possess basic knowledge on software process models,
3. Apply different software process models,
4. Analyze the cost of software development in practice, and
5. Evaluate and design software projects.

Course content: Software systems engineering, Ethical and professional responsibility, Organization, people and computer systems. Models of software processes, Iterative and approximate processes, Rational Unified Model, CASE. Management activities, Project planning, Scheduling activities, Risk management, Version planning, Version management, Software tools. Functional and non-functional requirements, User requirements, System requirements, Interface specification, Document of software requirements, Feasibility study, Analysis and validation of the requirements. System model and architectural design. Agile methods of software development. Reusability of software and testing.

Methods of teaching: 30 hours of lectures + 30 hours of lab exercises. Approximately 65 hours of personal study including home/group project assignments.

Grading System: Classroom Assessment 10%, Projects 40%, Final assessment 50 %

Literature:

1. Ian Sommerville, Software Engineering, 10th Edition 2016
2. Roger S. Pressman, Software Engineering, A Practitioner's Approach, 8th Edition 2014

Course title: Design and Analysis of Algorithms (Mandatory, Sem. V, 5 ECTS)

Lecturer: Prof. Asst. Dr. Avni Rexhepi

Course goal: The purpose of the course is to enable students to design algorithms and to do their detailed analysis, regarding efficiency, sustainability, time and space complexity and processor and memory requests. Algorithms for different problems will be defined in C#, C++ and/or Java.

Learning outcomes: On successful completion of the course, students will be able to design efficient algorithms based on main parameters for problem solution and to analyze algorithms in detail.

Course content: Algorithm design. Basic analysis of algorithms, input classes, space complexity. Searching and selecting algorithms, sequential search, binary search, selection. Sorting algorithms, design and analysis: insertion sort, bubble sort, selection sort, shell sort, radix sort, heap sort, merge sort, quick sort, polyphase merge sort. Numerical algorithms, calculating polynomials, matrix multiplication, linear equations. Matching algorithms, string matching, Knuth-Morris-Pratt algorithm, Boyer-Moore algorithm, approximate match algorithm. Graph algorithms, graph terminology, graph data structures, depth-first traversal, breadth-first traversal, minimum spanning tree, shortest-path first (Dijkstra's algorithm). Nondeterministic algorithms, NP problems. Other algorithmic techniques, approximation algorithms, TSP Problem, Bin-packing problem, back-pack problem, Graph-Coloring problem.
Methods of teaching: 30 hours of lectures + 30 hours of auditorial/lab exercises. Approximately 65 hours of personal study and exercise including home problems/tasks.
Grading System: Attendance **10%,** Seminar/Project 40%, Final Exam 50 %
Literature:

1. Jeffrey J. McConnell, "Analysis of Algorithms, An Active learning approach", Jones and Bartlett Publishers, ISBN: 0-7637-1634-0, 2001.
2. Steven Skiena, "The Algorithm Design Manual", Springer Verlag New York Inc., ISBN: 0-387-94860-0.

D. S. Malik, C++ Programming: Program Design Including, Data Structures, Course Technology, Thomson Learning, Boston, Massachusetts, ISBN 0-619-03569-2

Course title: Entrepreneurship and Innovation (Mandatory, Sem. 5, 4 ECTS)
Lecturer: Prof. Asst. Dr. Nora Sadiku

Course goal: The goal of the course is to equip students with a foundational understanding of entrepreneurship, developing skills for identifying, evaluating, and pursuing entrepreneurial opportunities.

Learning outcomes: Upon successful completion of this module, student will be able to:

1. Define fundamental entrepreneurship concepts and principles.
2. Explain the role of creativity, innovation, and risks in entrepreneurship
3. Differentiate various entry options, such as franchise and family business, in entrepreneurial endeavors.
4. Develop a comprehensive business plan for a hypothetical entrepreneurial venture
5. Propose different options for entering the entrepreneurial space

Course content: Introduction to entrepreneurship an innovation course, Concept of entrepreneurship, Nature of small business, Creativity and innovation, Risks and entrepreneurship, Business plan and entrepreneurship, Options for entering entrepreneurship, Franchise as entrepreneurial option, Family Business and succession, Entrepreneurial financing, Entrepreneurial marketing, Digital entrepreneurship, Corporate entrepreneurship, Social entrepreneurship and green entrepreneurship

Methods of teaching: 30 hours of lectures + 15 hours of laboratory exercises. Approximately 55 hours of personal study and exercise including seminars.
Grading System: Seminar 10%, Mid-term exams 40 %, Final Exam 50 %
Literature:

1. Veland Ramadani, Robert D. Hisrich, Nora Sadiku-Dushi & Shqipe Gërguri-Rashiti. Ndërmarrësia dhe menaxhimi i biznesit të vogël. Tetovë, 2022
2. John Bessant, "Entrepreneurship and innovation", John Wiley & Scons Inc. 2019
3. Charles Hampden-Turner, "Teaching Innovation and Entrepreneurship", Cambridge University Press 2009,

Course title: Microprocessors and microcontrollers (Elective, Sem. V, 5 ECTS)

Lecturer: Dr. Lavdim Kurtaj

Course content: Motivation and introduction to microprocessor systems. Logical circuit and microprocessor system design. Microprocessor systems and details about communication between units, time diagrams. Internal structure of 8051 and ARM, variations among different producers. Introduction to development system. Instruction set. Input/output ports and communication with peripherals. Programming in assembler and C/C++. Keyboard and display. Program structure. State diagrams. Timer function and programming. Interrupts - generation and processing. Serial communication. A/D and D/A conversion. Programming in higher level languages, Basic, C, C++.
Course objectives: Familiarity with parts of computer hardware. Uses of microprocessors and/ or microcontrollers in different applications. Basic microcontrollers concepts. Familiarity with architecture of 8051 and ARM family of microcontrollers. Familiarity with software development systems. Microcontroller programming in assembler, Basic, C, C++. Programming and utilization of peripheral devices.
Learning outcomes:
On successful completion of the course, students will be able to:

- know structure of microprocessor systems that are faced with;
- design microprocessor and microcontroller-based systems for specific application;
- write program for specific application;
- find and repair defects in microprocessor systems.

Teaching methodology: Combined lectures with simulations and demonstrations, discussions, laboratory exams, projects.
Evaluation methods: Intermediary evaluations 15%+15%, Project 40%, Final exam 15%+15%.
Concretization tools: Computer, projector, simulator, experimental development systems.
Ratio between the theoretical and practical part: 40:60
Literature

1. S. Mackenzie, The 8051 microcontroller, 4th Edition, Prentice-Hall, 2007
2. Muhammad Tahir and Kashif Javed, ARM Microprocessor Systems: Cortex-M Architecture, Programming, and Interfacing, CRC Pess, 2017
3. Renesas Synergy Development Kit, User's Manual, Renesas Electronics, 2015
4. D. V. Hall, Microprocessors and digital systems, McGraw-Hill
5. Muhammed Ali Mazidi, The 8051 Microcontroller And Embedded Systems Using Assembly and C, Pearson Education, 2007
6. Vinod G. Shelake, Rajanish K. Kamat, Jivan S. Parab, Gourish M. Naik, Exploring C for Microcontrollers: A Hands on Approach, Springer, 2007
7. Manuale të prodhuesëve për mikroprocesor dhe mikrokontroller

Course title: Internet Security (Elective, sem. V, 5 ECTS**)**

Lecturer: Prof. Dr. Blerim Rexha

The goal: Introducing the forms of attacks, algorithms for encryption / decryption, protocols for sending data in secure way through network, Firewalls, Viruses / Trojans, Wireless Security, IPsec.

Learning outcomes: On successful completion of the course students will be able to:

1. Understand and apply several forms of attacks, theft of passwords, social engineering, authentication failures, protocol failures, active and passive attackers.
2. Apply and analyze symmetric / asymmetric algorithm and to make the analysis of the safety of these algorithms.
3. Apply and analyze protocols for sending data in the safe mode, using digital certificates, Internet Mail Architecture.
4. Understand characteristics of firewall, types of firewalls, firewall location and configuration, proxy servers.
5. Analyze and evaluate security in wireless LAN, architecture models, WAP architecture.
6. Apply KALI Linux tools and explore top 10 vulnerabilities defined OWASP.

Course content: Introduction to Cryptography, Terminology, types of attacks. Symmetric algorithms: AES. Non-symmetric algorithms: Principles of Cryptosystems with Public Key, RSA, Diffie-Hellman, Elliptic Curve, Hash functions (one-way functions): MD5, SHA-1, SHA-256, Digital Signatures: Security in the Transport layer, Functionality of MIME and S / MIME, Architecture of Internet Mail, DKIM Strategy, E-mail Threats. Firewall Characteristics, Needs for firewall, Firewall configuration, Creation of a firewall to filter through a Linux PC, De-militarized Zone (DMZ), VPN. Introducing Network Components and Architecture Model, Services, Operation phases, Authentication phases, Key Management phases, Data transfer protection phases, WAP architecture, WAP End-to-End Security. Introduction to IP Security and benefits, Using KALI Linux, OWASP vulnerabilities.

Methods of teaching: 15 hours of lectures + 45 hours of lab exercises. Approximately 65 hours of personal study including two home/group project assignments.

Grading System: Classroom Assessment 10%, two home group assignments 90%

Literature:

1. Cryptography and Network Security, by William Stallings, ISBN 10:1-292-15858-1, published by Prentice Hall, 2017.
2. Internet Security, by Man Young Rhee, ISBN=0-470-85285-2, published by John Willey & Sons, 2003.
3. Penetration Testing: A Hands-On Introduction to Hacking Georgia Weidman, 2014
4. https://www.kali.org/kali-linux-documentation/ , 2019

Course title: Software Quality Assurance (Elective, Sem. V, 5 ECTS)

Lecturer: Prof. Assoc. Dr. Kadri Sylejmani

Course goal: This course provides in-depth knowledge about techniques for software testing, which aims at preparing students to successfully complete software development projects.

Learning outcomes: On successful completion of the course, students will be able to:
1. Design test cases for various levels of software testing that include unit testing, integration testing, system testing and acceptance testing; 2. Use techniques for black box testing, 3. Use techniques for white box testing, 4. Use various testing tools such as xUnit, NUnit, JUnit, PHPUnit, TestNG, etc.; 5. Stress and overload testing; 6. Perform analysis and static testing.

Course content: Software testing foundations. Software testing cycle. Unit testing. Integration testing. System testing. Acceptance testing. Testing software systems after addition of new modules. Static testing and analysis. Black box testing techniques. White box testing techniques. State transition testing. Stress and overload testing.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 65 hours of personal study and exercise including homework.

Grading System:
Attendance 10%, Practical project 30 %, Final Exam 60 %

Literature:

1. Software Testing Foundations. Second Edition, Andreas Spillner, Tilo Linz, and Hans Schaefer. Rocky Nook, Inc. 2007. ISBN 9781 9339 5208 6.
2. SOFTWARE TESTING Foundation Guide. Second Edition. Brian Hambling (Editor)
3. The Art of Software Testing. Second Edition. Glenford J. Myers, Software Testing and Quality Assurance Theory and Practice. Kshirsagar Naik

Course title: Data mining (Elective, Sem. V, 5 ECTS)

Lecturer: Prof. Dr. Lule Ahmedi

Course goal: Data mining is a relatively young but rapidly growing field that is concerned with developing techniques to, among others, assist businesses to make intelligent use of their repositories. For example, a supermarket might gather data on customer purchasing habits. Using data mining techniques, the supermarket can determine which products are frequently bought together and use this information when planning their business. This course will examine methods on recognizing patterns and relationships among available data and making predictions from an applications perspective. Experimentation with algorithms for data mining using a programming language will also be provided. Some often problems occurring during data mining will also be addressed.

Learning outcomes: Upon completion of this course, the student shall be able to:

1. Gain solid background on traditional approaches of data mining starting from data processing to description of existing data or prediction of unseen data using the algorithms in use of data mining.
2. Be capable of developing a basic data mining system.
3. Be able to first choose the right approach of data mining, and then address some common problems.

Course content: Topics include but are not restricted to: data preparation for mining, classification, association rules, clustering, overfitting, the imbalanced classes problem, anomaly detection.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 65 hours of personal study and exercise including project work.

Grading System:

Active attendance 10%, Project assignment 3 x 10% = 30%, Final Exam 60 %.

Literature:

1. Introduction to Data Mining (2nd Edition). Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar. Pearson, 2018.
2. Python Data Science Handbook: Essential Tools for Working with Data (1st Edition). Jake VanderPlas. O'Reilly Media, 2016.

Course title: Scripting Languages (Elective, Year III, Sem. V, 5 ECTS)

Lecturer: Prof. Asoc. Dr. Kadri Sylejmani

Course goal: To equip students with a comprehensive understanding of the design principles and practical applications of scripting languages, including design and development of libraries. This course aims to empower students with the knowledge and skills to proficiently manipulate text and data through subtle and complex coding techniques, enabling them to automate a diverse range of tasks.

Learning outcomes: Upon successful completion of this module, student will be able to:

1. Show examples of scripting languages and command interpreters with a specific emphasis on Python.
2. Identify good candidate administrative functions for scripting using Python.
3. Understand the benefits and challenges of scripting, particularly in the context of Python.
4. Demonstrate basic scripting in the Linux/Unix environment using Python.
5. Use regular expressions to search text files, specifically applying this skill within the context of Python.
6. Identify basic syntax and language elements of a Python script and write a complete Python script to automate common administrative functions.
7. Use the module mechanism to add functionality to a Python script.

Course content: This course encompasses the crucial aspect of creating new libraries as an integral component. Students will not only gain a profound understanding of scripting but also focus on the design and development of new libraries, emphasizing their importance in enhancing functionality and modularization within scripting languages. The curriculum delves into best practices for library creation, providing students with the skills to design, implement, and effectively utilize libraries to augment the capabilities of scripting languages, with a primary emphasis on Python. The course ensures that participants are well-equipped to contribute meaningfully to the expansion and optimization of scripting language libraries in practical applications.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 65 hours of individual study and exercises that include homework.

Grading System: Grading System: Activity in lectures and lab exercises 20%, Practical project 20 %, Final Exam 60 %

Literature:

1. Programming Language Pragmatics, 4th Edition, Michael Scott, ISBN-13: 978-0124104099, Morgan Kaufmann.
2. Programming and Problem solving with, PYTHON, 1st Edition, Ashok Namdev Kamthane and Amit Ashok Kamthane, McGraw Hill Education, 2016.

Course title: Data analysis (Elective, Sem. V, 5 ECTS)

Course goal: This course gives a comprehensive introduction to the methods and practices of computer-based data analysis. The course covers and intertwines four integral aspects of statistical analysis: data, statistical methods, mathematical foundations, and interpretation of results.

Learning outcomes: On successful completion of the course, students will be able to:

- Define main notions in the statistical data analysis;
- Explain mathematical backgrounds of main statistical procedures;
- Apply procedure of data preparation and visualization;
- Apply statistical test on real data;
- Analyze the relation between statistical variables by applying regression analysis and correlation analysis;
- Justify the adequacy of statistical inference for given data;
- Interpret the results of statistical data analysis and explain their practical meaning.

Course content: The course covers and intertwines four integral aspects of statistical analysis: data, statistical methods, mathematical foundations, and interpretation of results. The first part of the course gives an overview of statistical methods, approaches to data description, and data visualization and exploration methods. The second part is devoted to the foundations of statistical inference and covers the selection, application, and adequacy of parametric statistical tests for numeric and categorical data. The third part considers more advanced topics, such as non-parametric statistics, analysis of variance, and correlation analysis. All concepts are illustrated with examples and problem sets on real data in programming languages R and Python.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 60 hours of personal study and exercise including homework.

Grading System: Attendance 10%, Practical project 60 %, Final Exam 30 %

Literature:

- Ronald Walpole, Raymond Myers, Sharon Myers, Keying Ye (2012.), *Probability and Statistics for Engineers and Scientists*,
- David Diez, Christopher Barr, Mine Çetinkaya-Rundel (2015.), *OpenIntro Statistics*,
- Python Data Analysis: Perform data collection, data processing, wrangling, visualization, and model building using Python, Avinash Navlani, Amando Fandango, Ivan Idris, Packt Publishing, 978-1789955248, 2021
- Data Analytics, Data Visualization & Communicating Data: 3 books in 1: Learn the Processes of Data Analytics and Data Science, Create Engaging Data ... Present Data Effectively (All Things Data), Elizabeth Clarke, Kenneth M Fornari, ISBN: 978-1777967192

YEAR 3

SEMESTER 6

Course title: Professional Project (Mandatory, Sem. VI, 6 ECTS)
Lecturer: Every professor in department

Course goal: This course is a comprehensive exploration of the various subjects in computer and software engineering. Its primary objective is to provide students with the skills and knowledge necessary to develop complex applied projects as part of a team. To achieve this, students will be required to draw on the knowledge and skills acquired during their previous five semesters of study. Through a combination of lectures, practical assignments, and group projects, students will learn how to collaborate effectively and use the latest software engineering techniques and tools to build sophisticated programs. By the end of the course, students will be equipped with a deep understanding of programming concepts and a range of practical skills that will enable them to develop high-quality software applications that meet real-world needs.

Learning outcomes: Upon successful completion of the project, students will be able to:

- Collaborate effectively with their team members to develop a computer and software product that meets the requirements and standards of the project. They will learn how to use collaboration tools such as Git, Slack, Trello, and Jira to communicate and manage tasks and deliverables.
- Create comprehensive documentation for the project that covers all aspects of the development process, including the project scope, requirements, design, testing, and maintenance. They will learn how to use documentation tools such as Confluence and Google Docs to create and share project documentation.
- Integrate and design libraries and APIs to implement various functionalities of the program and ensure modularity, scalability, and maintainability. They will learn how to use programming languages such as Java, Python, or JavaScript and frameworks such as Spring or React to implement the project functionalities.
- Verify and validate the program by employing various testing techniques and methodologies, such as unit testing, integration testing, and system testing, to ensure its correctness, reliability, and performance. They will learn how to use testing tools such as JUnit, Mockito, and Selenium to test the project.

Teaching Methods: The course will consist of 15 hours of lectures, 15 hours of coaching, and 30 hours of laboratory exercises, in addition to approximately 50 hours of personal study using video lectures and e-learning materials. Furthermore, students will be required to work collaboratively with their peers for approximately 50 hours on group assignments and projects.

Grading System: Attendance 20%, Practical project 80%

Literature:

- **Kernighan, Brian W. y Pike, Rob. The Practice of Programming. Addison-Wesley.ISBN:020161586X.1999.Cat:INF/681.3.06/KER**
- Pressman, Roger.S. Software engineering a practitioner's approach, 6ªed. McGraw Hill Higher Education, 2005. Cat.INF/681.3.06/PRE.
- Cormen, Thomas H.et al. Introduction to algorithms. The MIT press. ISBN: 0262531968.2001. Cat: INF/510.5/COR.

Course title: Internship (Mandatory, Semester VI, 6 ECTS)

Course goal: The students gain professional experience in the field of computer and software engineering in one of the local or international companies/institutions under supervision from an expert from the field.

Learning outcomes: To be qualified for professional work in the field of computer and software engineering, hence become better prepared for the labor market.

Course Content: The content of this course depends on the company where the student shall finish 120 working hours. As a result, the content is drafted jointly from the coordinator of professional internships, appointed from the company, and the student who is going to work in such company. The coordinator of professional internships, who is appointed from the company, guides the student throughout the duration of his/her work in the company, and, at the end, he confirms the successful completion of the students' internship by signing the exam grade sheet.

Methods of teaching: 120 hours work in a company and for writing the seminar.

Grading System: Seminar Work 20%, Defense 80%. TOTAL: 100%.

Course title: Distributed systems (Mandatory, Sem. VI, 6 ECTS)

Lecturer: Prof. Dr. Isak Shabani

Course goal: The aim of this course is to enable and prepare students to understand and apply distributed system techniques.

Learning outcomes: On successful completion of the course, students will be able to:

1. To understand main concepts and system model of distributed;
2. To compare distributed system;
3. To creates distributed systems;
4. Manage distributed objects;
5. To realize e project relating a particular issue with distributed systems.

Course content: Characterization of distributed systems. System models. Networking and internetworking in Distributed Systems. Interprocess Communications. Remote Invocation. Indirect Communication. Distributed components and objects. Web Services. Peer-to-peer systems. Distributed File System. Name Services. Global Time and State. Distributed Transactions. Distributed Replication. Web based distribution systems.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 65 hours of personal study and exercise including homework.

Grading System:
Attendance 10%, Practical project 30 %, Final Exam 60 %

Literature:

1. George Coulouris, Jean Dollimore, Tim Kindberg, Peter Baer Galvin dhe Greg Gagne, "Distributed Systems – Concepts and Design", 5th Edition, 2012.
2. Andrew S. Tanenbaum, "Distributed Systems – Concepts and Paradigms", 2rd Edition, 2007.
3. Manish Varshnev and Shanoo Agarwal, "Concepts of distributed system", 2016

Course title: Cloud Computing (Elective, Sem. VI, 6 ECTS)

Lecturer: Prof. Asst. Dr. Artan Mazrekaj

Course goal: The purpose of the course is to equip students with basic knowledge of virtualization and cloud integration services. Knowing and understanding the platforms and communications in the cloud environment. Efficient use of hardware resources, assessment of the performance of the Cloud resources.

Learning outcomes: On successful completion of the course, students will be able to:

1. To understand the key concepts and structure of the cloud.
2. To compare and analyze the methods and algorithms for cloud exploitation.
3. To configure and build a cloud platform by familiarizing in depth with service-oriented architecture.
4. To understand the role and capabilities of cloud services integration.
5. To configure platforms in order to have an efficient use of resources.

Contents: Introduction to Cloud Computing. Types of Cloud Computing. Cloud Computing Models. Virtualization, Infrastructure as a Service- IaaS. Platform as Service - PaaS. Software as Service - SaaS. Services Oriented Architecture (SOA). Migration of Cloud Services. Managing the SLA. Cloud Resource Allocation. Introduction to big data management. Cloud Security. Cloud Challenges. Case studies with different frameworks. Analysis and study on cloud computing platforms, using of simulators, etc.

Methods of teaching: Lectures, laboratory exercises/simulations. Personal study by students and independent project work.
Grading System: Attendance 10%, Seminar paper 10%, Project 20 %, Final Exam 60 %
Literature:

1. Thomas Erl, "Cloud Computing: Concepts, Technology, Security and Architecture", , Eric Barcelo, second edition, Pearson, 2023.
2. Dan C. Marinescu, "Cloud Computing: Theory and Practice", third edition, Elsevier-MK, 2022
3. Edited by: Rajkumar Buyya, James Broberg, Andrzej Goscinski, "Cloud Computing: Principles and Paradigms", Wiley, 2011.

Course title: Fundamentals of Artificial Intelligence (Elective, Sem. VI, 6 ECTS)

Lecturer: Prof. Asst. Dr. Avni Rexhepi

Course goal: Introduce students to the basic concepts, techniques, and applications of Artificial Intelligence (AI) and enable them to understand and apply AI tools and techniques to solve real-world problems.

Learning outcomes: On successful completion of the course, students will be able to:

- Understand the basic concepts, goals, and limitations of Artificial Intelligence,
- Understand the basic principles of problem-solving using search and reasoning,
- Understand the basic concepts and techniques of Machine Learning,
- Understand the basic concepts and applications of Natural Language Processing,
- Apply AI techniques to solve real-world problems.

Course content: Introduction to AI basic concepts and applications (what it is and how it works), problem-solving methods and algorithm development using search and reasoning. Introduction to Machine Learning and Natural Language Processing. Techniques for supervised and unsupervised learning, clustering, classification, and prediction. Laboratory exercises using Python programming language.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 90 hours of individual study and exercises that include homework.

Grading System: Activity in lectures and lab exercises 10%, Practical project 40 %, Final Exam 50 %

Literature:

- Artificial Intelligence: A Modern Approach, 4th Edition, Stuart Russell, Peter Norvig. ISBN-13: 978-0134610999, ISBN-10: 0134610997, Prentice Hall.
- Python Machine Learning, 3rd Edition, Sebastian Raschka, Vahid Mirjalili. ISBN-13: 978-1801078356, ISBN-10: 1801078352, Packt Publishing.
- Natural Language Processing with Python, Steven Bird, Ewan Klein, and Edward Loper. ISBN-13: 978-0596516499, ISBN-10: 0596516495, O'Reilly Media.

Course title: Game programming (Elective, Sem. VI, 6 ECTS)

Lecturer: Prof. Dr. Blerim Rexha

Course goal: The Game Programming course aims to equip students with a comprehensive understanding of the principles and practices involved in developing interactive and engaging computer games. Students will delve into both theoretical concepts and practical skills necessary for game development, fostering creativity and critical thinking in the process.

Learning outcomes: On successful completion of the course, students will be able to:

1. Recall and articulate fundamental concepts of game design, programming languages, and development tools.
2. Demonstrate a solid comprehension of the theoretical underpinnings of game development, including game mechanics, physics, and artificial intelligence.
3. Apply programming skills to create functional prototypes and simple games, integrating essential game development techniques.
4. Evaluate and analyze existing games, identifying successful design patterns, and critically assessing the impact of programming decisions on user experience.
5. Develop a complete, polished game project, showcasing creativity, problem-solving, and proficiency in game programming concepts.

Contents: Overview of the game development process, Introduction to game engines and development environments, Basic principles of game design, Review of programming languages (e.g., C++, Java, Python) in the context of game development, Data structures and algorithms relevant to game programming. Fundamentals of 2D and 3D graphics, Animation techniques and principles, Introduction to shaders and visual effects. Newtonian physics in games, Collision detection and response, Implementing realistic motion and interactions. Basics of game AI algorithms, Pathfinding and decision-making for non-player characters (NPCs), Implementing adaptive and learning AI. Work on the final game project

Methods of teaching: Lectures, laboratory exercises/simulations. Personal study by students and independent project work.

Grading System: Attendance 10%, Seminar paper 10%, Project 20 %, Final Exam 60 %

Literature:

1. Eric Matthes, “Python Crash Course”, second edition, Pearson, 2019.
2. Robert Nystrom, “Game Programming Patterns”, 1st edition, Genever Benning, 2014
3. Ian Millington and John Funge, “Artificial Intelligence for Games”, Morgan Kaufmann, 2021.

Course title: Augmented, Virtual and Mixed Reality (Elective, Sem. VI, 6 ECTS)

Lecturer: Prof. Dr. Lule Ahmedi

Course goal: Learn the basic concepts of AR, VR and XR. Understand what mixed reality offers in real-world apps. Develop applicable 3D virtual environments.

Learning outcomes: On successful completion of the course, students will be able to:

- Know the basic concept and framework of virtual reality
- Define mixed reality and differentiate between virtual reality and augmented reality
- Understand what mixed reality offers in real-world apps
- Address the need for and use of virtual and mixed reality technologies in the world today
- Develop 3D virtual environments
- Develop 3D interaction techniques and immersive virtual reality applications
- Use the technology for managing large scale VR environments in real time

Course content: Teach students core concepts of VR, AR and XR, to develop 3D virtual environments, develop 3D interaction techniques and immersive virtual reality applications, demonstrate ways to mix real and virtual objects and environments.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 60 hours of personal study and exercise including homework.

Grading System: Attendance 10%, Practical project 60 %, Final Exam 30 %

Literature:

- Virtual, Augmented, and Mixed Realities in Education (by Dejian Liu (Editor), Chris Dede (Editor), Ronghuai Huang (Editor), John Richards (Editor))
- Augmented Reality and Virtual Reality: The Power of AR and VR for Business (by M. Claudia tom Dieck (Editor), Timothy Jung (Editor))
- Creating Augmented and Virtual Realities: Theory and Practice for Next-Generation Spatial Computing (by by Erin Pangilinan (Author), Steve Lukas (Author), Vasanth Mohan (Author))
- Digital Anatomy: Applications of Virtual, Mixed and Augmented Reality (Human–Computer Interaction Series) (by Jean-François Uhl (Editor), Joaquim Jorge (Editor), Daniel Simões Lopes (Editor), Pedro F. Campos (Editor))
- Complete Virtual Reality and Augmented Reality Development with Unity: Leverage the power of Unity and become a pro at creating mixed reality application (by Jesse Glover (Author), Jonathan Linowes (Author))

Course title: Data engineering (Elective, Sem. VI, 6 ECTS)

Lecturer: Prof. Dr. Lule Ahmedi

Course goal: This course covers data engineering in their diversity on models, physical storage, distribution, as well as on their access, integration, or as well manipulation through queries.

Learning outcomes: Upon completion of this course, the student shall be able to:

1. Demonstrate familiarity with advanced data models from object-relational (objects in SQL), semi-structured model (XML), noSQL, distributed in cloud, and in OLAP cubes.
2. Model and manipulate data in whatever of these models through advanced queries in SQL, Xquery, or in noSQL.
3. Develop information systems based on any of these models.
4. Compare and integrate data of these distinct models.
5. Use and get familiar indirectly with new technologies, like Oracle DBMS, programming in Java, and XML and noSQL data manipulation systems.

Course content: A preliminary list of topics cover (mainly based on the audience): 1. Object-relational databases. Advanced SQL. Optimal physical organization of databases. 2. XML semi-structured data model: DTD and XML Schema, XQuery. 3. noSQL data model. Queries in noSQL. 4. Distributed (cloud) data. Distributed transaction management. Replication. Distribute queries. 5. Data Warehousing and OLAP. 6. Access to data. Comparison of models. Data integration. Trends and perspectives in the future.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 65 hours of personal study and exercise including project work.

Grading System:
Active attendance 10%, Project assignment 3 x 10% = 30%, Final Exam 60 %.

Literature:

1. Principles of Database Management: The Practical Guide to Storing, Managing and Analyzing Big and Small Data (1st Edition). Wilfried Lemahieu, Bart Baesens, Seppe vanden Broucke. Cambridge University Press. 2018.
2. Database System Concepts (6th Edition). Abraham Silberschatz, Henry F. Korth, S. Sudarshan. McGraw-Hill Education, 2010.
3. Database Concepts (8th Edition). David M. Kroenke, David J. Auer, Scott L. Vandenberg, Robert C. YoderPearson, Pearson, 2018.

Course title: Platform-Based Development (Elective, Sem. VI, 6 ECTS)

Lecturer: Prof. Dr. Lule Ahmedi

Course goal: Teach students to develop and implement programming tasks via platform-specific APIs and present the results to a group of peers.

Learning outcomes: On successful completion of the course, students will be able to:

- Describe the difference between platform-based and general purpose programming and application development.
- Understand the different concepts, characteristics, and implementation mechanisms on various platforms,
- Design and develop applications using various platforms,
- Design a user friendly interface,
- Integrate multiple applications on different platforms.

Course content: Overview of platforms (Web, Mobile, Game, Industrial etc.). Programming via platform-specific APIs. Overview of Platform Languages (Objective C, HTML5, etc.). Programming under platform constraints. Mobile platform constraints. Emerging Technologies. Domain Specific Languages. Industrial Platform Constraints. Game Platform Constraints.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 40 hours of personal study and exercise including homework.

Grading System: Attendance 10%, Practical project 40 %, Final Exam 50 %

Literature:

- Internet and World Wide Web: How To Program. Fifth Edition, Paul Deitel, Harvey Deitel, Abbey Deitel. Pearson. 2012. ISBN-13: 978-0132151009.
- Test-Driven Development with Python. First Edition, Harry Percival. O'Reilly, 2014. ISBN-13: 978-1491958704.
- Head First, Android Development: A Brain-Friendly Guide. Second Edition, Dawn Griffiths and David Griffiths. O'Reilly. 2017. ISBN-13: 978-1491974056.

Course title: Fundamentals of Blockchain Technologies (Elective, Sem. VI, 6 ECTS)

Lecturer: Prof. Dr. Blerim Rexha

Course goal: Introduce students to blockchain technologies that allows a set of actors to maintain a distributed record of transactions in a shared data storage environment in a way that is verifiable and permanent.

Learning outcomes: On successful completion of the course, students will be able to:

- Understand the structure of a Blockchain.
- Analyze the incentive structure in a blockchain based system and critically assess its functions, benefits and vulnerabilities.
- Evaluate the setting where a blockchain based structure may be applied, its potential and its limitations.
- Attain awareness of the new challenges that exist in monetizing businesses around blockchains and smart contracts.
- Describe and understand the differences between the most prominent blockchain structures and permissioned blockchain service providers, as well as rising alliances and networks.

Course content: Intro to cryptography & cryptocurrencies. Bitcoin nuts and bolts. Wallets: managing and protecting crypto assets. Consensus: network models, corruption tolerance, sybil resistance. Nakamoto Consensus: security, attacks and incentives. Large Scale Consensus: Availability/Finality, Randomness Beacons, VDFs. Ethereum: Decentralized Apps, EVM, and the Ethereum blockchain. Programming in solidity. Stablecoins and oracles. Decentralized exchanges. Lending systems. Legal aspects and regulation.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises. Approximately 60 hours of personal study and exercise including homework.

Grading System: Attendance 10%, Practical project 40 %, Final Exam 50 %

Literature:

- Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, ISBN-13: 978-0691171692, ISBN-10: 0691171696
- The Internet of Money: A collection of talks by Andreas M. Antonopoulos Paperback, Andreas M. Antonopoulos
- The Age of Cryptocurrency: How Bitcoin and the Blockchain Are Challenging the Global Economic Order Paperback, Paul Vigna, Michael J. Casey

Course title: Biometrics & Forensics (Elective, Sem. VI, 6 ECTS)

Lecturer: Prof. Dr. Blerim Rexha

The goal: To provide students with actual biometric technologies utilized by personal documents, actual standards and regulations as well as cybernetic crimes and their detection and protection from them.

Learning outcomes: On completion of this course, students will be able to:

1. Apply biometric technologies for digital identity and modern symmetric encryption algorithms,
2. Apply and understand the latest legal and technical standards on identification documents,
3. Apply public key algorithms, use and understand digital signatures and public key infrastructure,
4. Have knowledge about latest cyber-attacks,
5. Use different forensic tools, and
6. Analyze and evaluate document digital proves

Course content: Will include: Concepts and trends in document identity, Biometric properties, legal and technical standards, ICAO and IEC, Analysis of Fingerprints and Fingermarks, Automated Fingerprint Identification Systems: From Fingerprints to Fingermarks, Face Recognition Technologies for Evidential Evaluation of Video Traces, Secure communication protocols, EU experience in eID security, Computer forensics, Preparing the environment, Evidence gathering, Data extraction, Encryption and password policy, Forensic tools and final reports..

Methods of teaching: 30 hours of lectures + 30 hours of lab exercises. Approximately 65 hours of personal study including home/group project assignments.

Grading System: Classroom Assessment 10%, Projects 40%, Final assessment 50 %

Literature:

Essential:

1. Editors: Massimo Tistarelli, Christophe Champod: Handbook of Biometrics for Forensic Science, 2017
2. Walter Fumy, Manfred Paeschke, Handbook of eID Security , ISBN=078-3-89578-379-1, 2011
3. Digital Forensics with Open Source Tools. Cory Altheide and Harlan Carvey, ISBN: 978-1-59749-586-8, Elsevier publication, April 2011

Course title: Introduction to Large Language Models (Elective, Sem. VI, 6 ECTS)

Lecturer: Prof. Dr. Blerim Rexha

The goal: This course gives a comprehensive introduction to the methods and practices of language models. The course covers and intertwines integral aspects of large language models and use: text to text transformation, deep conceptualization, generative pre-training, supervised, unsupervised, semi-supervised models and so on.

Learning outcomes: On completion of this course, students will be able to:

1. Define the fundamental concepts associated with large language models, including key terms such as natural language processing, neural networks, and pre-training.
2. Explain the underlying architecture and working principles of large language models, illustrating how neural networks process and generate human-like language.
3. Implement and fine-tune a pre-trained language model for specific natural language processing tasks, demonstrating proficiency in model adaptation and problem-solving in various domains.
4. Evaluate the ethical implications and societal impacts of large language models, considering issues such as bias, privacy, and responsible AI deployment.

Course content: This course aims to cover cutting-edge research topics centering around pre-trained language models. We will discuss their technical foundations (BERT, GPT, T5 models, mixture-of-expert models, retrieval-based models), emerging capabilities (knowledge, reasoning, few-shot learning, in-context learning), fine-tuning and adaptation, system design, as well as security and ethics. In the first weeks, students will delve into foundational concepts with modules such as 'Introduction to Natural Language Processing' and 'Neural Networks Basics.' Following this, the course will progress to understanding large language models with chapters like 'Architecture of Large Language Models' and 'Training Mechanisms.' Weeks 5-8 will focus on practical applications through 'Hands-On Implementation' sessions where students will apply learned concepts. The middle part of the course will explore ethical considerations with chapters such as 'Ethical Issues in AI' and 'Societal Impacts of Large Language Models.' The final weeks will challenge students to showcase their understanding and creativity in 'Project Development,' where they will create and present their own applications. Throughout the course, regular assessments and exercises will reinforce learning and allow for continuous improvement, ensuring students leave with a comprehensive understanding of large language models and their applications in the realm of computer and software engineering.

Methods of teaching: 30 hours of lectures + 30 hours of lab exercises. Approximately 65 hours of personal study including home/group project assignments.

Grading System: Classroom Assessment 10%, Projects 40%, Final assessment 50 %

Literature:

Essential:

1. Jay Alammar, Maarten Grootendorst (2024), Hands-On Large Language Models, Publisher(s): O'Reilly Media, Inc. ISBN: 9781098150969

2. Thimira Amaratunga, Understanding Large Language Models: Learning Their Underlying Concepts and Technologies 1st ed. Edition (2023)
3. Richard A Aragon, Large Language Models: A New Frontier in Artificial Intelligence Paperback (2023).

Course title: Coding & Algorithms Bootcamp (Elective, Year III, Sem. VI, 6 ECTS)

Lecturer: Prof. Asoc. Dr. Kadri Sylejmani

Course goal: The overarching aim of this course is to elevate students current programming proficiency by cultivating a profound mastery of essential algorithms and coding skills, thereby enhancing their capabilities as a student within the given timeframe of the program. Specific emphasis will be placed on the enhancement of algorithmic problem-solving skills, the improvement of the ability to plan and describe solutions to problems, the further development of the ability to translate thoughts into code and explain that code to others, with the ultimate goal of maximizing the chances of superior performance in any coding interview.

Learning outcomes: On successful completion of the course, students will be able to:

- Proficiency in solving complex problems using complex algorithms.
- Accurate planning and clear articulation of solutions.
- Elevated coding skills translating abstract thoughts into efficient code.
- Articulate and effective explanation of code to others.
- Strategic analysis of coding challenges for optimal solutions.
- Comprehensive readiness for interviews through understanding algorithms and effective problem-solving.

Course content: This comprehensive course delves into the intricacies of programming and algorithmic problem-solving. Beginning with an overview of the course structure, the focus swiftly shifts to fundamental concepts like Strings and Arrays, paving the way for a deeper understanding of computational complexity. Java is explored for String Manipulation and Regular Expressions, equipping students with practical coding skills. Interview Techniques, Testing, and Defensive Programming strategies are addressed to enhance coding proficiency and promote robust solutions. The course covers diverse problem-solving approaches, from mathematical problem-solving to counting and memoization techniques. Combinatoric and randomized techniques are explored, preparing students for real-world challenges. Practical applications include problem simulation, game strategies, and the implementation of State Machines, Bit Manipulation, and the Sliding Window technique. The course concludes with a comprehensive review and exam preparation, ensuring students emerge with a profound mastery of algorithms and coding skills, ready for coding interviews and real-world scenarios.

Methods of teaching: 30 hours of lectures + 30 hours of laboratory exercises.
Approximately 65 hours of individual study and exercises that include homework.

Literature:

- Cracking the Coding Interview: 189 Programming Questions and Solutions, 6th Edition, Gayle Laakmann McDowell, ISBN13: 978-0984782857, Barnes & Noble
- Grokking Algorithms: An illustrated guide for programmers and other curious people, Aditya Bhargava, ISBN13: 978-1617292231, MIT Press.
- System Design Interview – An insider's guide, Alex Xu, ISBN13: 979-8664653403

Course title: Diploma Thesis (Mandatory, Semester VI, 6 ECTS)
Course goal: Diploma thesis is a comprehensive and independent assignment, where student must demonstrate the capability to analyze the problem from theoretical or practical aspect., in order to provide a solution by applying acquired knowledge from different courses and literature.
Learning outcomes: On successful completion of this module student will be able:

- To gain confidence on acquired knowledge
- To get skills for further study from literature
- To consult with the supervisor
- To present their work in a written form, and standard language
- To present their findings in an oral evaluation

Course Content: Diploma thesis topic may be proposed by the supervisor or be chosen by the student. It must be compatible with the student's qualification profile.
Methods of teaching: Determined by the regulation for diploma thesis of the faculty.

Literature:

1. Depending on the topic of the diploma thesis, literature will be provided by the supervisor.

1. Practical applications of biomedical engineering;
2. Sensors and instrumentation, types of biosensors used for collecting data in the domain;
3. Collecting, annotating and processing data from biosensors and images, building a pipeline of machine learning models to infer and predict