

BSc Computer Science

Year I						
Semester I			Hour/Week			
Nr	O/E	Course	L	E	ECTS	Professor
1	O	Elementary mathematics	2	2	6	Mr. Sc. Ramadan Limani
2	O	Discrete mathematics	2	2	6	Dr. Sc. Qëndrim Gashi
3	O	Programming and algorithms	2	2	6	Dr. Sc. Faton Berisha
4	O	Introduction to computer science	2	2	6	Dr. Sc. Eliot Bytyçi
5	Z	English language I (for Comp. Sci)	2	2	6	Dean's office
6	Z	Academic writing	2	2	6	Dean's office
Total semester I			10	10	30	
Semester II			Hour/Week			
Nr	O/E	Course	L	E	ECTS	Professor
1	O	Calculus I	3	2	6	Dr. Sc. Naim Braha
2	O	Linear algebra and analytic geometry	2	2	6	Dr. Sc. Armend Shabani
3	O	Data structures	2	2	6	Dr. Sc. Eliot Bytyçi
4	O	Computer networks and WWW	2	2	6	Dr. Sc. Artan Berisha
5	Z	Algebra	2	2	6	Dr. Sc. Menderes Gashi
6	Z	Computer applications I	2	2	6	Dr. Sc. Armend Shabani
7	Z	English language II	2	2	6	Dean's office
Total semester II			11	10	30	
Viti II						
Semester III			Hour/Week			
Nr	O/E	Course	L	E	ECTS	Professor
1	O	Calculus II	3	2	6	Dr. Sc. Naim Braha
2	O	Numerical analysis	3	3	7	Dr. Sc. Faton Berisha
3	O	Computer architecture	2	2	6	Dr. Sc. Ermir Rogova
4	O	Databases	2	2	6	Dr. Sc. Ermir Rogova
5	Z	Operating systems	2	2	5	Dr. Sc. Ermir Rogova
6	Z	Programming for mobile devices	2	2	5	Dr. Sc. Artan Berisha
7	Z	Introduction to Cloud computing	2	2	5	Dr. Sc. Korab Rrmoku
8	Z	Computer applications II	2	2	5	Dr. Sc. Artan Berisha
Total semester III			12	11	30	
Semester IV			Hour/Week			
Nr	O/E	Course	L	E	ECTS	Professor
1	O	Artificial intelligence	2	2	6	Dr. Sc. Eliot Bytyçi
2	O	Probability theory	2	2	6	Dr. Sc. Bujar Fejzullahu
3	O	OOP and GUI programming	2	2	6	Dr. Sc. Faton Berisha
4	O	Data security	2	2	6	Dr. Sc. Artan Berisha
5	Z	UX / UI Design	2	2	6	Dr. Sc. Korab Rrmoku
6	Z	Matrix theory	2	2	6	Dr. Sc. Muhib Lohaj
7	Z	Computing theory	2	2	6	Dr. Sc. Faton Berisha
8	Z	Linear programming	2	2	6	Dr. Sc. Menderes Gashi
Total semester IV			10	10	30	
Viti III						
Semester V			Hour/Week			
Nr	O/E	Course	L	E	ECTS	Professor
1	O	Programming for www	2	2	6	Dr. Sc. Ermir Rogova
2	O	Analysis of algorithms	2	2	6	Dr. Sc. Elver Bajrami

3	O	Statistics	2	2	6	Dr. Sc. Edmond Aliaga
4	O	Distributed programming	2	2	6	Dr. Sc. Artan Berisha
5	Z	Accounting	2	2	6	Dr. Sc. Ejup Fejza
6	Z	Internet of Things (IoT)	2	2	6	Dr. Sc. Korab Rrmoku
7	Z	Game development	2	2	6	Dr. Sc. Artan Berisha
8	Z	Game theory	2	2	6	Dr. Sc. Qëndrim Gashi
9	Z	Information systems	2	2	6	Dr. Sc. Ermir Rogova
Total semester V			10	10	30	
Semester VI			Hour/Week			
Nr	O/E	Course	L	E	ECTS	Professor
1	O	Project (industrial practice)	2	2	6	Department
2	O	Image processing and computer graphics	2	2	6	Dr. Sc. Artan Berisha
3	O	Software engineering	2	2	6	Dr. Sc. Ermir Rogova
4	O	Data analysis	2	2	6	Dr. Sc. Eliot Bytyçi
5	Z	Programming for www server	2	2	6	Dr. Sc. Ermir Rogova
6	Z	Software development in OOP	2	2	6	Dr. Sc. Korab Rrmoku
7	Z	Parallel programming	2	2	6	Dr. Sc. Artan Berisha
8	Z	Stochastic processes	2	2	6	Dr. Sc. Elver Bajrami
Total semester VI			10	10	30	

Short description of courses

Semester I

Course: Elementary Mathematics

ECTS:6

Short content: Powers, roots and their properties. Fundamental algebraic identities and binomial formula. Some characteristic functions (absolute value function, signum function, floor and ceiling function, fractional function and Dirichlet function). Linear equations (linear inequalities) in one variable, and their applications. Quadratic equation and quadratic mapping, quadratic inequalities, and their applications to business and economics. Exponential mappings, exponential equation and inequalities and their applications. Logarithms and their properties. Logarithmic functions and their graphs. Logarithmic equations and inequalities and their applications. Plane trigonometry (trigonometric circle, basic trigonometric identities, addition trigonometric identities, trigonometric equations and inequalities, trigonometric and their inverse trigonometric functions, sine and cosine theorems and their applications).

Course objectives: The main aim of this course is to introduce some basic mathematics to the students of first year of study, so they can be able to easily use these basic mathematical notions to other mathematics or computer science courses that are related to mathematics.

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

- Easily use the mathematics notions from this course to other courses and in their real life;
- Define main mathematical concepts used in this course by offering examples and counterexamples;

- Apply the notions from this course to convert different real life problems into a mathematical problem (model), solve them mathematically, and interpret the obtained solutions;
- Analyse the complexity of an algorithm by studying the worst case, average case and best case;
- Clearly express their ideas, orally and in written form while solving a mathematical or real life problem.

Teaching and learning methods: Lectures, discussions, exercises, tests and final exam.

Assesment: Regular attendance 10%; first test 20%; second test 20%; (written exam 40%); final exam 50%; total 100%. In case a student does not pass the exam according to the above policy, then he or she will have to take the written and oral exams. The written exam weighs 40%, while the oral one weighs 60%. Final grading will be according to the following table:

Number of points out of 100	Grade
0-49	5 (fail)
50-59	6 (six)
60-69	7 (seven)
70-79	8 (eight)
80-89	9 (nine)
90-100	10 (ten)

Necessary and additional tools: Whiteboard, markers, eraser, computer and LCD projector.

Proportion between theory and exercise is 1:1.

Literature :

- R. Limani: Kursi i Matematikës elementare, 2002.
- Terry H. Wesner, Harry L. Nustad: Intermediate Algebra with applications, WCB Group 1991.

Student workload			
Activity	Hours	Days/weeks	Total
Lectures	2	15	30
Exercise sessions (with TA)	2	15	30
Practical work			
Office hours	1	15	15
Fieldwork			
Midterms, seminars	2	10	20
Homework			
Self-study	3	15	45
Final exam preparation	2	5	10
Projects, presentations, etc.			
Total			150

Course: Discrete Mathematics

ECTS: 6

Short description: This course covers basic notions of discrete mathematics for computer science, including sets, maps, relations, counting techniques and graphs.

Mathematical logic, sets and functions. Integers. Algorithms. Matrices. Mathematical thinking. Induction. Recursion. Counting techniques. Permutations and combinations. Discrete probability theory. Relations. Graphs. Graph properties and applications. Trees. Boolean algebras.

Goals: The course aims to help students create the mathematical foundation in order to understand the fundamental principles of computer science, to apply mathematical notions in computer science and to advance their critical and creative thinking.

Expected outcomes: At the end of the course students should be able to describe, explain and apply the notions of sets, maps, relations, counting techniques and graphs. They will model problems from computer science and real life based on the notions studied in this course.

Teaching and learning methods: Lectures, discussions, exercises, consultations, homework, tests and exam.

Assessment:

Participation (10%); Midterms (2 x 15%); Homework (15%); Projects (15%); Final exam (30%).

Grading schema:

Points	Grade
0-49	5
50-60	6
61-70	7
71-80	8
81-90	9
91-100	10

Necessary and additional tools: Whiteboard markers, eraser, whiteboard, computer and LCD projector.

Proportion between theory and practice is 2:2

Literature:

- K. H. Rosen, *Discrete Mathematics and Its Applications*, 7th Ed.; Mc-Graw Hill, 2011 (or later editions)
- Q. Gashi, *Shënime nga matematike diskrete*, Prishtinë, 2022 (lecture notes in Albanian)

Student Workload			
Activity	Hours	Days/weeks	Total
Lectures	2	15	30
Exercise sessions	2	15	30
Office hours	1	15	15
Midterms, seminars	2	2	4
Self-study	3	15	45
Projects, presentations, etc.	3	4	12
Time spent in exams	4		4
Final exam preparation	10		10
Total			150

Course title: Programming and Algorithms

ECTS: 6

Course description: This course covers parts of algorithm building and basic programming in Java programming language.

Course objectives: Providing students with basic knowledge of object-oriented programming, component structure, and control structure in Java. Ability to program simple Java applications according to the model-view-controller architecture.

Expected learning outcomes: Upon successful completion of the course, students will be able to:

- Able to implement MVC architecture to encode simple Java application designs;
- Be able to apply knowledge of the basic principles of OOP, component structure and control structure to code simple Java applications;
- Be able to apply basic algorithmic templates to solve different problem situations.

Teaching methodology: Lectures, Exercises, Quizzes, Laboratory Work, Laboratory Work, discussions, mid-term exams, final exam.

Evaluation methods and criteria: Periodic Exam: 25%, Laboratory work: 15%, Regular line: 10%, Final exam 50%.

Concretization tools: Laptops, smart screens, computers in laboratories, markers, pencils.

Relation between the theoretical and practical part of the study: 2:2.

Literature:

- Schmidt, Programming principles in Java: architectures and interfaces, Kansas State University, 2003
- J. Schildt, Java 2: The complete reference, McGraw-Hill, 2001.
- Horton, Beginning Java 2, Wrox Press, 2000.
- P. S. Wang, Java with object-oriented programming, Brooks/Cole-Thomson Learning, 2003.

Student Workload (should be in compliance with student's Learning Outcomes)			
Activity	Hours	Day/ Week	Total
Lectures	2	15	30
Theory/ Lab Work/Exercises	2	15	30
Practical Work			
Consultations with the teacher	2	2	4
Field Work	1	15	15
Test, seminar paper			
Homework			
Self-study (library or home)			
Preparation for final exam	4	15	60
Assessment time (test, quiz, final exam)			8
Projects, presentations, assignments, etc.			3
Total			150

Course title: Introduction to computer science

ECTS: 6

Course description: This initial course with interdisciplinary content is oriented to prepare students with the basic concepts in the field of computer science with a special emphasis on the form of storage of notes on the computer, data manipulation (processor and main memory), systems architecture operational and explanation of basic knowledge with computer networks and the Internet. In the exercises, students will be introduced to the basic concepts of the Python programming language.

Course objectives: The course has as an objective to teach to students' main principles of computer science and to be coherent with the new upcoming principles from the field.

Expected learning outcomes: Upon the completion of this course, students will be able to:

- have basic knowledge of computer science and be trained to distinguish the relevance and interconnection of special courses in this field.
- create a culture of permanent tracking of trends in the field of computer science.
- gain basic knowledge about the form of computer data storage.
- to know the forms of communication of devices within the computer.
- to understand the importance of human - computer communication,
- as well as gain knowledge of the Python programming language.

Teaching methodology: Interactive learning, lectures, laboratory exercises, seminar papers, consultations.

Evaluation methods and criteria: Periodic exam 1 (30%), Project preparation and defense (20%), Regular attendance and engagement in class (10%), Periodic exam 2 (30%). Those students who do not pass the course with periodical exams, undergo the final exam which is evaluated with 60%.

Grading

Points	Grade
0-49	5
50-60	6
61-70	7
71-80	8
81-90	9
91-100	10

Concretization tools: Laptops, smart screens, computers in laboratories, markers, pencils.

Relation between the theoretical and practical part of the study: 2:2.

Literature

- J. Glenn Brookshear, *Computer Science-An Overview*, Addison Wesley, latest edition
- Python Org Tutorial <https://docs.python.org/3/tutorial/>

Contribution on student load (must correspond with learning outcomes)			
Activity	Hours	Days/week	Total
Lectures	2	15	30
Exercise laboratory	2	15	30
Contact with lecturer/consultations	1	10	10
Mid-terms, seminars	2	3	6
Individual time spent studying (at the library or home)			75
Total			151

Course: English language I (for computer science)

ECTS: 6

Course description: With the development of English as a global and computer language, as well as the needs to communicate in this language, it has become a necessary requirement that would prepare future students who study Computer Science, to be ready to understand, communicate and write when there is a need to exchange ideas with other students of this field intercontinentally. The subjects include different texts with content in the field of computer science as well as different real-life situations in this field.

Course objectives: Speaking is the central part of the philosophy of learning a foreign language, therefore also of the English language, and through this course students will practice this skill more. Training students in speaking, in pairs and in groups. Enabling students to talk about real-life situations in the field of computer science.

Expected learning outcomes: After completing this course students should be able to:

- know how to read texts related to computers;
- Communicate about things that happen in their real life in their field;
- Understand and use some of the many words of computer science;
- Know how to understand what others speak in English, so they will be prepared to listen to English;
- Make comments about what they have heard from the media, newspapers, people.

Teaching methodology: Lectures, exercises, discussions.

Evaluation methods and criteria: Essay (30%), Seminar Paper (20%), Regular Attendance (10%), Final Exam (40%), Total (100%)

Assessment: 50 points to 59 points – 6; 60 points to 69 points – grade 7; 70 points to 79 points – grade 8; 80 points to 89 points – grade 9; 90 points to 100 points – grade 10.

Concretization tools/ IT: Marker, sponge, board.

The ratio between the theoretical and practical part of the study: The ratio between the theoretical and practical part is: 2 hours of theory and 2 hours of practice.

Literature:

- Santiago Remacha Esteras, Infotech- English for computer users 3rd Edition, Student's book, 2009

Student workload			
Activity	Hours	Days/weeks	Total
Lectures	2	15	30
Exercise sessions (with TA)	2	15	30
Office hours	1	15	15
Midterms, seminars	20	1	20
Homework	5	1	5
Self-study	2	15	30
Final exam preparation	10	1	10
Time spent on assessment (tests, quizzes, final exam)	5	1	5
Projects, presentations, etc.	5	1	5

Total			150
--------------	--	--	------------

Course: Academic writing

ECTS: 6

Course description: During this course, students will learn and practice the process and methods of academic writing based on the step-by-step technique. The ability to write clearly and accurately is a necessary condition that must be learned. Every individual, regardless of habits or previous knowledge, has the opportunity to learn to write well and similar to other habits. This requires practice, preparation, organization, concentration and experience.

Course objectives: The course will help students to write, express and think at an academic level. Students are prepared to enter their dissertation, which will be professionally completed after the student has achieved the objectives of the academic writing course.

Expected learning outcomes: After completing this course students should be able to:

- Develop critical thinking and perform critical readings of literature;
- Acquire (learn) sufficient knowledge and skills to write academically;
- Know how to place the thesis within an argumentative text;
- Analyze and evaluate the strength of the written argument;
- Know how to systematize ideas and borrowed text fragments according to international citation systems.

Teaching methodology: Lectures, exercises, work in seminar groups, discussions, etc.

Evaluation methods and criteria: Regular attendance (5%), Seminar work (25%), First test (15%), Second test (15%), Final exam (40%), Total (100%)

Concretization tools/ IT: Marker, sponge, board, computer, projector.

The ratio between the theoretical and practical part of the study: The ratio between the theoretical and practical part is: 2 hours of theory and 2 hours of practice.

Literature:

- Bernard Zeneli, Sonila Danaj, *Shkrim Akademik*, Tiranë, 2006.
- Kathleen McMillan&Janathan Weyers(përkthyes: Majlinda Nishku dhe Xhevat Lloshi), *Si të shkruajmë ese dhe detyra universitare*, Tiranë, 2009.

Student workload			
Activity	Hours	Days/weeks	Total
Lectures	2	15	30
Exercise sessions (with TA)	2	15	30
Midterms, seminars	1	15	15
Homework	2	15	30
Self-study	2	15	30
Final exam preparation	1	15	15
Time spent on assessment (tests, quizzes, final exam)			
Projects, presentations, etc.			
Total			150

Semester II

Course title: Calculus I

ECTS : 6

Course content: This course covers basic notions from mathematical analysis, such as sets of real and complex numbers, countable sets, meaning of number series, limits of series, number of, number series, limits of functions, continuous functions, derivatives of functions, the application of derivatives of functions in the graphical representation of functions, etc.

Objectives of the course: In this course, the main goal is for students to understand the fundamental notions of calculations, with special emphasis on the practical application of basic concepts from countable sets, limits of sequences, numerical series, etc.

Learning outcomes: After completing this course students should be able to:

- They have knowledge on numerical communities, in particular on the community of real and complex numbers;
- Know the basic concepts about ranges and their limits;
- Know the basic classes of functions and their properties;
- Understand the properties of continuous functions;
- Solve problems related to the derivative of the function and its application in different problems, such as in the graphical presentation of functions

Teaching methodology: Lectures, theoretical and numerical exercises, discussions, group work.

Assessment methods and passing criteria: First Test (50%), Second Test (50%) or Final Exam (100%)

Assessment: 50 points to 59 points – 6; 60 points to 69 points – grade 7; 70 points to 79 points – grade 8; 80 points to 89 points – grade 9; 90 points to 100 points – grade 10.

Means of concretization/ IT: Marker, pencil, board, computer, projector.

The ratio between the theoretical and practical part of the study: The ratio between the theoretical and practical part is: 2 hours of theory and 2 hours of practice.

Literature:

- N. Braha and I. Shehu, Mathematical Analysis I, Prishtina, 2006
- Brian S. Thomson, Judith B. Bruckner, Andrew M. Bruckner: Elementary real analysis, Thomson - Bruckner, 2001
- Walter Rudin, Principles of mathematical analysis, McGraw-Hill, 1976

Contribution to the student's workload (which must correspond to the student's learning outcomes)			
Activity	hour	Day/week	in total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30
Practical work			
Contacts with the teacher/consultations	1	15	15
Field exercises			
Colloquiums, seminars	10	1	10
Homework	5	1	5

Student's independent study time (in the library or at home)	2	15	30
Final exam preparation	20	1	20
Time spent on assessment (tests, quizzes, final exam)	5	1	5
Projects, presentations, etc.	5	1	5
TOTAL			150

Subject: Linear algebra with analytic geometry

ECTS: 6

Course content : Systems of linear equations. Matrices. Matrix operations. Matrices. Their implementation. Determinants and their properties. Application of determinants. Personal values. Vector Spaces. Subspaces. Basis and dimension of vector space. Applications of vector spaces. Spaces with internal production. Orthonormal bases. Gram-Shmid process. Linear transformations. Linear transformation matrix. Application of linear transformations. The equation of the plane. Justice in space. Straight line and plane

Objectives of the course: Acquaintance of students with the basic concepts and ideas of linear algebra and analytic geometry as well as with their application in other mathematical disciplines.

Expected outcomes: After completing this course (subject), the student is able to:

- Models and solves systems of linear equations using various methods
- Apply matrix algebra to solving various problems
- Defines the transformation matrix from a basis B to the basis B' in the space R^n
- Apply inner product properties to solving problems in plane and space geometry
- Implements Gram-Shmidit process for orthogonality

Forms of teaching and learning: Lecture, discussion, homework, use of audiovisual tools.

Assessment methods: Active participation (10%), Homework (10%), First colloquium (20%), Second colloquium (20%), Final exam (40%)

Grading: Grading is taken according to this scheme

MARKS	Grading
0-49	5
50-60	6
61-70	7
71-80	8
81-90	9
91-100	10

Means of concretization: Marker, pencil, ordinary board, digital board, computer, projector.

The ratio between the theoretical part and exercises is 1:1

Main literature:

- R. Larson, DC Falvo, *Elementary Linear Algebra*, Houghton Mifflin Harcourt Publishing Company, 2009.
- G. Williams, *Linear algebra with applications*, ninth edition, Jones & Bartlett Learning, 2019.

Additional literature:

- Berani, *Analytical geometry, ETMM*, Pristina, 1988

- Hoxha, *Mathematics I (Summary of solved tasks)* , Prishtina, 2002

Contribution to the student's workload			
Activity	Lessons	Day/Week	in total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30
Consultations with the teacher	1	15	15
Preparation for periodical tests	2	15	30
Periodic tests	2	2	4
Home work	2	15	30
Individual learning (in the library or at home)	3	3	9
Final exam	2	1	2
Total			150

Course title: Data structures

ECTS: 6

Short content: This course is aimed at preparing students with basic knowledge of data structures in general and with a focus on exercises in the Java programming language. In addition to data structures, recursion and some applications of search algorithms will be examined.

Purpose: The primary purpose of the course is to introduce students to data structures and their applications.

Expected results: Upon successful completion of this course, students will be able to implement the array as the most basic data structure; to implement and distinguish the benefits of using recursion: to become familiar with some of the other data structures such as lists, linked lists, stack and queue structures, maps, hash tables, logs.

Forms of teaching and learning: nteractive learning, lectures, laboratory exercises, seminar papers, consultations.

Evaluation methods: Periodic exam (35%), Project preparation and defense (20%), Regular attendance and engagement in class (10%), Final exam (35%).

Grading

Points	Grade
0-49	5
50-60	6
61-70	7
71-80	8
81-90	9
91-100	10

Concretization tools: Laptops, smart screens, computers in laboratories, markers, pencils.

Relation between the theoretical and practical part of the study: 2:2.

Literature:

- Y. Daniel Liang, Introduction to Java Programming and Data Structures, Pearson, Latest edition
- Goodrich, M. T., Tamassia, R., Goldwasser, M. H. Data Structures and Algorithms in Java. Wiley, Latest edition

Contribution on student load (must correspond with learning outcomes)			
Activity	Hours	Days/week	Total
Lectures	2	15	30
Exercise laboratory	2	15	30
Contact with lecturer/consultations	1	10	10
Mid-terms, seminars	2	3	6
Individual time spent studying (at the library or home)			75
Total			151

Course title: Computer Networks and WWW

ECTS: 6

Short description: Introduction to communication networks. Network architectures, classification and topology. Communication channel and information packet. Communication protocols, layered models: Open System Interconnection Reference Model (OSI RM), Internet model. IP protocol and other network layer protocols in Internet. Organization of IP-based networks. Transport layer protocols, TCP and UDP. Host naming, Internet domains. Information and multimedia services and Internet applications. Local area network, wide area network, internetworking.

Course aim and expected results: Students will gain fundamental knowledge of communication networks, network architecture and protocols, with special emphasis on local area networks, Internet and public networks. Students will gain knowledge and skills enabling them understanding of communication network design problems and professional approach to them. They will develop learning skills necessary to continue to undertake further study of communication networks.

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

- define concept, architecture and organization of communication networks,
- explain how communication networks operate and their functionality,
- apply knowledge about communication networks and protocols, analyze protocol functions and services, as well as protocol stacks in order to select appropriate ones, analyze organization of public and private networks based on IP protocol,
- define security threats and available solutions in the Internet, design network models including local area networks, Internet subnetworks and Internet Access

Teaching and learning forms: Lectures, exercises, quizzes, laboratory work, discussions, periodic exams, final exam.

Evaluation methods:

Hands-on work 30%,

First midterm 35%,

Second midterm 35%,

Final exam 70% (if midterm results were not satisfactory)

Points from laboratory work and colloquiums are collected and evaluated according to the following criteria:

Grading: Grading is done according to this scheme:

Point	Grade
<50	5
>=50 and <60	6
>=60 and <70	7
>-70 and <80	8
>=80 and <90	9
>=90	10

Means of concretization / IT: Laptop, projector, computers in the hall, markers, erasers, UTP cable, switch, router.

The relationship between theoretical and practical part of the study 2:2

Literature:

- James F. Kurose, Keith W. Ross (2017.), Computer Networking: A Top-Down Approach Featuring the Internet, 7/e, Pearson
- Andrew S. Tanenbaum, David J. Wetherall (2010.), Computer Networks, 5/e, Prentice Hall
- Cisco Networking Academy: Cisco networking curriculum (CCNA 1)

Contribution on student load (must correspond with learning outcomes)			
Activity	Hours	Days/week	Total
Lectures	2	15	30
Exercise laboratory	2	15	30
Practice work			
Contact with lecturer/consultations	1	15	15
Field exercises			
Mid-terms, seminars	2	10	20
Homework			
Individual time spent studying (at the library or home)	2	15	30
Final preparation for the exam	1	10	10
Quizzes	1	10	10
Projects, presentations, etc.	1	5	5
Total			150

Course: Algebra

ECTS: 6

Short content: This course introduces the basic concepts of abstract algebra, including the number systems, groups, rings, fields, and integral domains. There is an emphasis on the use of sets as a basis for defining and working with groups, rings, fields, and integral domains.

Course objectives: To develop a working knowledge of the elementary definitions, examples, and theorems of groups, rings, and fields.

Learning outcomes: Upon completing this course students should be able to:

- Classify groups, rings, integral domains, and fields.
- Apply Lagrange's Theorem to analyze the structure of groups.

- Relate normal subgroups, homomorphisms, and factor groups.
- Apply algebraic concepts in different programming techniques.

Teaching and learning methods: Lectures, discussions, exercises, homework, tests and final exam.

Method of evaluating outcomes and grading policy (assessment): Homework: 5%; regular attendance: 5%; first test 20%; second test 20%; (written exam 50%); final exam 50%; total 100%. Final grading: 50%-59% (six); 60%-69% (seven); 70%-79% (eight); 80%-89% (nine) and 90%-100% (ten).

Necessary and additional tools: Whiteboard, markers, eraser, computer and LCD projector.

Proportion between theory and exercise is 2:2.

Literature:

- Emrush Gashi: Kursi i algebrës së lartë, Prishtinë, 1998
- A First Course in Abstract Algebra By J. Fraleigh

Student workload – Semester II			
Activity	Hours	Days/weeks	Total
Lectures	2	15	30
Exercise sessions (with TA)	2	15	30
Practical work	-	-	-
Office hours	1	15	15
Fieldwork	-	-	-
Midterms, seminars	2	2	4
Homework	2	15	30
Self-study	2	15	30
Final exam preparatin	5	1	5
Time spent in exams	2	3	6
Projects, presentations, etc	-	-	-
Total			150 hours

Subject: Computer applications I

ECTS: 6

Brief content : This course provides an introduction to technical writing, rendering complex figures and presentations with LaTeX, and the use of the various packages that LaTeX offers. The course offers techniques for writing documents, preparing presentations, constructing figures within documents, etc. A special importance will be given to the annotation of documents containing mathematical formulas.

Purpose of the subject

Overview with the basic elements in LaTeX with the aim of processing various professional and academic documents.

Expected results: After completing this course (subject), the student is able to:

- Process various notes in LaTeX
- Process documents containing mathematical formulas

- Draw and import various figures within LaTeX
- Use different packages;
- They prepare materials according to different scientific formats.

Forms of teaching and learning: Lecture, discussion, homework, use of audiovisual tools.

Assessment methods: Active participation (10%), Project presentation (20%), First test (30%), Final exam (40%).

Grading: Grading is taken according to this scheme

MARKS	Grading
0-49	5
50-60	6
61-70	7
71-80	8
81-90	9
91-100	10

Means of concretization: Marker, pencil, ordinary board, digital board, computer, projector.

The ratio between the theoretical part and exercises is 1:1

Main literature:

- S. Kottwitz, *LaTeX Beginner's Guide*, Second Edition, Packt Publishing Ltd., 2021.
- G. Grätzer, *More Math Into LATEX Fifth Edition*, Springer, 2016.

Additional literature:

- M. Öchsner, A. Öchsner, *Advanced LaTeX in Academia Applications in Research and Education*, Springer, 2021.

Contribution to the student's workload			
Activity	Lessons	Day/Week	total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30
Consultations with the teacher	1	15	15
Preparation for periodic tests and for the seminar project	2	15	30
Periodic tests	2	2	4
Home work	2	15	30
Individual learning	3	3	9
Final exam	2	1	2
Total			150

Subject : English language II

ECTS : 6

Course content: This course is a continuation of the English Language I course. The course includes various texts with content in the field of computer science as well as various real-life situations in this field. Students will have the opportunity to do a mini research on different topics of this field in English.

Objectives of the course: Continually become proficient in reading and writing the English language.

Learning outcomes: After completing this course students should be able to:

- Be able to comment on any read text;
- Be able to communicate in groups;
- They become proficient in speaking, in pairs and in groups;
- Be able to talk about real-life situations in the field of computer science;
- Become proficient in the use of vocabulary in the field of computer science.

Teaching methodology: Lectures, exercises, discussions.

Evaluation methods and passing criteria: Essay (30%), Seminar paper (20%), Regular attendance (10%), Final exam (40%), Total (100%)

Assessment: 50 points to 59 points – 6; 60 points to 69 points – grade 7; 70 points to 79 points – grade 8; 80 points to 89 points – grade 9; 90 points to 100 points – grade 10.

Concretization tools/ IT: Marker, pencil, board.

The ratio between the theoretical and practical part of the study: The ratio between the theoretical and practical part is: 2 hours of theory and 2 hours of practice.

Literature:

- Santiago Remacha Esteras, Infotech- English for computer users 3rd Edition, Student's book, 2009

Contribution to the student's workload (which must correspond to the student's learning outcomes)			
Activity	hour	Day/week	in total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30
Practical work			
Contacts with the teacher/consultations	1	15	15
Field exercises			
Colloquiums, seminars	20	1	20
Homework	5	1	5
Student's independent study time (in the library or at home)	2	15	30
Final exam preparation	10	1	10
Time spent on assessment (tests, quizzes, final exam)	5	1	5
Projects, presentations, etc.	5	1	5
TOTAL			150

Semester III

Subject : Calculus II

ECTS : 6

Course content: This course covers the topics on the meanings of integrals, primitive function, basic properties of indefinite integrals, methods of calculating integrals, method of substitution, method of integration by parts, integration of rational and irrational functions, integration of trigonometric functions, integration of any functions, the meaning of the definite integral and its properties, classes of integrable functions, applications of the definite integral, surfaces, lengths of curves, volume of rotating bodies, surface area of rotating bodies, functional series, polynomial series and decomposition of functions into polynomial series.

Objectives of the course: In this course, the main goal is for students to understand the fundamental notions that emerge from mathematical analysis, with special emphasis on the practical application of concepts on functions, integrals and series.

Learning outcomes: After completing this course students should be able to:

- Be equipped with sufficient knowledge on indefinite and definite integrals;
- They know how to apply certain integrals in concrete calculations related to parts of programming;
- Understand the importance of integrable functions and their properties;
- They are equipped with basic knowledge about series and decompositions of functions in series;
- They know how to apply different series in programming for calculations in the form of cycles.

Teaching methodology: Lectures, theoretical and numerical exercises, discussions, group work.

Assessment methods and passing criteria: First Test (50%), Second Test (50%) or Final Exam (100%)

Assessment: 50 points to 59 points – 6; 60 points to 69 points – grade 7; 70 points to 79 points – grade 8; 80 points to 89 points – grade 9; 90 points to 100 points – grade 10.

Means of concretization/ IT: Marker, pencil, board, computer, projector.

The ratio between the theoretical and practical part of the study: The ratio between the theoretical and practical part is: 2 hours of theory and 2 hours of practice.

Literature:

- Brian S. Thomson, Judith B. Bruckner, Andrew M. Bruckner: Elementary real analysis, Thomson - Bruckner, 2001
- Walter Rudin, Principles of mathematical analysis, McGraw-Hill, 1976
- Adnadevic.D and Kadelburg .Z, Mathematical Analysis I (Serbian) Belgrade, 1998

Contribution to the student's workload (which must correspond to the student's learning outcomes)			
Activity	hour	Day/week	in total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30
Practical work			
Contacts with the teacher/consultations	1	15	15
Field exercises			
Colloquiums, seminars	20	1	20
Homework	5	1	5
Student's independent study time (in the library or at home)	2	15	30

Final exam preparation	10	1	10
Time spent on assessment (tests, quizzes, final exam)	5	1	5
Projects, presentations, etc.	5	1	5
TOTAL			150

Subject: Numerical analysis

ECTS: 7

Course content: An introductory course of numerical analysis, with emphasis on coding the algorithms of numerical methods.

Objectives of the course: Enabling students to apply methods of numerical analysis to problem solving from different applications. Training in software implementation of numerical methods algorithms.

Expected outcome: Upon successful completion of the course, students will be able to:

- Understand the relationship between computer arithmetic and error propagation
- Apply iterative methods to numerically solve an equation with a single variable
- Apply direct or iterative methods for numerical solution of a system of linear and nonlinear equations
- Identify the advantages and disadvantages of different numerical methods
- Implement algorithms and implement numerical software.

Teaching methodology: Lectures, numerical exercises, laboratory exercises, laboratory works.

Assessment methods and passing criteria:

Periodic Exam: 30%

Laboratory work: 15%

Attendance: 10%

Final exam 45%

Total 100%

MARKS	Grading
< 51	5
51 to 60	6
61 to 70	7
71 to 80	8
81 to 90	9
> 90	10

Means of concretization: Marker, eraser, board, computer room, projector.

The ratio between the theoretical part and exercises The ratio between the theoretical and practical part is: 2 hours of theory and 2 hours of practice

Literature:

- RL Burden, JD Faires, *Numerical analysis*, Brooks/Cole, 2001
- CF Gerald, PO Wheatley, *Applied numerical analysis*, Addison-Wesley, 1994
- D. Schmidt, *Programming principles in Java: architectures and interfaces*, Kansas State University, 2003

Student Workload (should be in compliance with student's Learning Outcomes)			
Activity	Hours	Day/ Week	Total
Lectures	3	15	45

Theory/ Lab Work/Exercises	3	15	45
Practical Work			
Consultations with the teacher	2	2	4
Field Work			
Test, seminar paper	5	1	5
Homework	1	15	15
Self-study (library or home)	3	15	45
Preparation for final exam	5	2	10
Assessment time (test, quiz, final exam)	3	2	6
Projects, presentations, assignments, etc.			
Total			175

Course : Computer architecture

ECTS : 6

Course content: this course includes basic notions of computer architecture and digital logic, von Neuman models, MARIE machine, symbol codes, input and output operational units.

Objectives of the course: Familiarity with the architecture of contemporary computer systems and the connection between them system software and computer architecture.

Learning outcomes: After completing this course students should be able to:

- Understand the internal structure of computer operations and functions
- Understand the specifics of a modern computer system and make the right decisions when making comparisons;
- Write simple programs coded in assembler;
- Explain the principles of designing and developing computer systems;
- Understand how system software works and how it relates to computer architecture

Forms of teaching and learning: Lectures, discussions, exercises, consultations, homework, tests and exams.

Assessment methods: Practical work: 30%, First test: 20%, Second test: 20%, Third test: 20%, Final exam: 10%.

Grading scheme:

points	Grading
0-49	5
50-61	6
62-73	7
74-85	8
86-97	9
98-100	10

Means of concretization: digital board, computer, projector.

The ratio between the theoretical and practical part is: 1:1

Literature:

- Linda Null, Julia Lobur: *The Essentials of Computer Organization and Architecture* , Jones and Bartlett Publishers Inc., 2018
- William Stallings: *Computer Organization and Architecture*, Global Edition, 2015
- Brian R. Hall and Kevin J. Slonka: *Assembly Programming and Computer Architecture for Software Engineers*

Contribution to the student's workload			
Activity	Hour	Day/week	Total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30
Contacts with the teacher/consultations	1	15	15
Colloquiums, seminars	1	6	6
Student's independent study time (in the library or at home)	4	15	60
Projects, presentations	9	1	9
Total load hours			150

Course: Databases

ECTS: 6

Short description: Processing and knowledge of basic database elements, data presentation formats, relational databases, relational algebra, basic and advanced knowledge in SQL, stored procedures, functions, database normalization, semi-structured databases (XML, JSON).

Course objectives: Relational database design, knowledge of DDL orders, DML orders, data types, subqueries, relational algebra, database normalization, basic JSON and XML elements, transactions, query by example, logical and arithmetic operators, JDBC, ODBCs.

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

- *Know how to design a database*
- *To know how to use elements of relational algebra*
- *Know to use orders in SQL*
- *Make database normalization*
- *Make database connection using the Java / JDBC programming language*
- *Know how to build JSON and XML database and*
- *Build questionnaires in XPath, XQuery, and JSONPath*

Teaching and learning methods: Lectures, discussions, hands-on exercises, consultations, homework, tests and exam.

Assesment: *Hands-on work: 30%, First test: 20%, Second test: 20%, Third test: 20%, Final exam: 10%.*

points	Grading
0-49	5
50-61	6
62-73	7
74-85	8
86-97	9
98-100	10

Necessary and additional tools

Whiteboard markers, eraser, whiteboard, computer and LCD projector.

Proportion between theory and practice is 1:1

Basic Literature:

- *Database Systems: Design, Implementation, & Management 13th edition, Carlos Coronel, Steven Morris, Cengage Learning, 2018*

Additional Literature:

- *Database System Concepts, Avi Silberschatz, Henry F. Korth, S. Sudarshan, McGraw Hill, 2019*
- *Essential SQL on SQL Server 2008, Sigha Bagui, Richard Earp, Jones and Bartlett, 2011*
- *Database Management Systems, Raghu Ramakrishnan and Johannes Gehrke, McGraw-Hill, 2007*

Student Workload			
Activity	Hours	Days/weeks	Total
Lectures	2	15	30
Exercise sessions	2	15	30
Office hours	1	15	15
Midterms, seminars	1	6	6
Self-study	4	15	60
Projects, presentations, etc.	9	1	9
Total			150

Course : Operating systems

ECTS : 5

Course content: Enables students to successfully apply the concepts of operating systems. Knowledge of the construction and operation of computer systems, the roles and services of operating systems as intermediaries between user programs on the one hand and computer system hardware on the other.

Objectives of the course: The purpose of the course is to provide students with modern knowledge in illustrating the functions of Operating Systems with concrete examples, comparative knowledge in Windows, Unix, Linux, Minix and other Operating Systems. Basic knowledge and understanding of the workings of operating systems.

Learning outcomes: After completing this course students should be able to:

- To describe and identify the main concepts and structure of operating systems .
- Compare and contrast operating systems.
- Manage process conflicts and threads of execution .
- Manipulate memory, processor and input /output units .
- Analyze and review operating systems.

Forms of teaching and learning: Lectures, discussions, exercises, consultations, homework, tests and exams.

Assessment methods: Practical work: 30%, First test: 20%, Second test: 20%, Third test: 20%, Final exam: 10%.

Grading scheme:

points	Grading
0-49	5
50-61	6
62-73	7

74-85	8
86-97	9
98-100	10

Means of concretization: digital board, computer, projector.

The ratio between the theoretical and practical part is: 1:1

Literature:

- Abraham Silberschatz: *Operating System Concepts, 10th Edition, 2018*
- Andrew S. Tanenbaum: *Modern Operating Systems, 4th Edition, 2014*

Contribution to the student's workload			
Activity	Hour	Day/week	in total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30
Contacts with the teacher/consultations	1	10	10
Colloquiums, seminars	1	6	6
Student's independent study time (in the library or at home)	3	15	45
Projects, presentations	4	1	4
Total load hours			125

Course title: Programming for mobile devices

ECTS 5

Short description: The course is designed to offer students an opportunity to acquire knowledge needed to develop Android applications, and also server-side services which will be used by these applications. Topics and themes covered include: foundations of the Kotlin programming language, object-oriented programming in Kotlin, Android platform architecture, graphical user interface, using specific hardware, data access for Android, accessing different services on Internet, developing RESTful web services and accessing them from Android.

Course aim and expected results: Upon completion of the course the student should be able to: employ object-oriented programming principles, use common Java classes and interfaces, apply data structures and streams, use Java logging and exception handling facilities, access databases from Java, use Eclipse integrated development environment, independently develop Android applications, utilize Android GUI components, access specific Android hardware, independently develop distributed Android applications and effectively work in small teams.

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

- develop application in object-oriented programming language Kotlin independently and in team
- develop application for device with operating system Android
- design and develop service on server
- apply network programming in development of Android applications
- relate service with database on Android and on server

Teaching and learning forms; Lectures, discussions, exercises, consultations, homework, colloquiums and exams.

Evaluation methods

Programming tasks: 15 point,
Seminar: 15 point,
First test 35 point,
Second test 35 point.

Grading: Grading is done according to this scheme

Point	Grade
<50	5
>=50 and <60	6
>=60 and <70	7
>=70 and <80	8
>=80 and <90	9
>=90	10

Points from programming tasks and seminar work that are achieved by the student during the course will be valid for each term. So the student is always subject to the exam with a maximum of 70 points.

Means of concretization / IT: Blackboard, digital board, computer, projector.

The relationship between theoretical and practical part of the study 2:2

Literature:

- Android Programming with Kotlin for Beginners, John Horton, 2019, Packt
- Android Development with Kotlin, Marcin Moskala, Igor Wojda, 2017, Packt

Contribution on student load (must correspond with learning outcomes)			
Activity	Hours	Days/week	Total
Lectures	2	15	30
Exercise laboratory	2	15	30
Practice work			
Contact with lecturer/consultations	1	15	15
Field exercises			
Mid-terms, seminars	1	10	10
Homework			
Individual time spent studying (at the library or home)	2	15	30
Final preparation for the exam	1	10	10
Quizzes			
Projects, presentations, etc.			
Total			125

Course: Introduction to Cloud Computing

ECTS: 5

Course Description: This course introduces students to the concept of cloud computing and its key technologies. Students will learn about the benefits and challenges of cloud computing, cloud service

models, deployment models, and key technologies such as virtualization, containers, and orchestration. The course will also cover cloud security, cloud management, and governance.

General Competencies: Students will gain a fundamental understanding of cloud computing, including its benefits, challenges, and key technologies. Students will develop skills in deploying, managing, and securing cloud-based services, and will be able to identify appropriate use cases for different cloud service models.

Learning Outcomes: Upon completion of this course, students will be able to:

- Explain the concept of cloud computing, including the benefits and challenges of using cloud-based services.
- Identify the different cloud service models and deployment models and understand their differences.
- Deploy and manage cloud-based services using virtualization, containers, and orchestration tools.
- Understand cloud security and apply appropriate security measures to protect cloud-based services.
- Understand cloud management and governance and apply best practices to manage cloud-based services.

Forms of Teaching: Lectures, Exercises, Quizzes, Laboratory Work, Discussions, Mid-term exams, Final exam

Grading Method: Attendance and class activity (10%), Semesterly practical project (30%), Lab work (15%), Final exam (45%)

Grades

piket	Nota
0-49	5
50-60	6
61—70	7
71-80	8
81-90	9
91-100	10

IT Concretization Tools: Cloud platforms such as AWS, Azure, or GCP, virtual machines, containers, orchestration tools, programming languages such as Python, and related software tools.

Report between theoretical and practical lecture hours: 2:2

Literature:

- Thomas Erl, Ricardo Puttini, Zaigham Mahmood (2013), Cloud Computing: Concepts, Technology & Architecture, Prentice Hall
- Frank Leymann, Dieter Roller, Markus Klems, Slinger Jansen (2013), Cloud Computing: Principles and Paradigms, Wiley
- Cloud Academy: Cloud computing training and certification courses

Contribution to the student's workload			
Activity	Hours	Day/Week	Total
Lectures	2	15	30

Theoretical/laboratory exercises	2	15	30
Contacts with the teacher/consultations	2	2	4
Colloquiums, seminars	2	2	4
Student's independent study time (in the library or at home)			57
Total load hours			125

Course title: Computer applications II

ECTS: 5

Short description: Learning computer applications for further study in computer science as well as solving many mathematical problems through application software.

Course aim and expected results: Basic Matlab programming, object-oriented programming, 2D and 3D graphic data presentation, data analysis, reading and processing images, solving mathematical problems from linear algebra, analysis, optimization methods. After completing the course, the student will be able to:

- To import and export data,
- To know how to program in MatLab,
- To know how to create M-File functions,
- Know to visualize the data,
- Analyze data based on basic methods.

Teaching and learning forms: This course is explained through lectures, numerical exercises, assignments and seminar work (where it is foreseen to be accomplished with continuous escort of the assistant and lecturer of the subject).

Evaluation methods

Programming tasks: 20 point,
Seminar: 10 point,
Test 1: 35 point,
Testi 2: 35 point.

Grading: Grading is done according to this scheme

Point	Grade
<50	5
>=50 and <60	6
>=60 and <70	7
>-70 and <80	8
>=80 and <90	9
>=90	10

Points from programming tasks and seminar work that are achieved by the student during the course will be valid for each term. So the student is always subject to the exam with a maximum of 70 points.

Means of concretization / IT: Laptop, computer, projector, table.

The relationship between theoretical and practical part of the study 2:2

Literature :

- Introduction to MATLAB® for Engineers, William J. Palm III, McGraw Hill 2009
- Official web page with latest version of MatLab documentation:
https://www.mathworks.com/help/pdf_doc/matlab/index.html?s_tid=mwa_osa_a

MATLAB® Primer

MATLAB® Desktop Tools and Development Environment

MATLAB® Data Import and Export

MATLAB® Mathematics

MATLAB® Data Analysis

MATLAB® Programming Fundamentals

MATLAB® Object-Oriented Programming

MATLAB® Graphics

MATLAB® 3-D Visualization

Contribution on student load (must correspond with learning outcomes)			
Activity	Hours	Days/week	Total
Lectures	2	15	30
Exercise laboratory	2	15	30
Practice work			
Contact with lecturer/consultations	1	15	15
Field exercises			
Mid-terms, seminars	1	10	10
Homework			
Individual time spent studying (at the library or home)	2	15	30
Final preparation for the exam	1	10	10
Time spent in evaluation (tests, final exam)			
Projects, presentations, etc.			
Total			125

Semester IV

Subject: Artificial Intelligence

ECTS: 6

Short content: In this course, fundamental principles for solving real-life problems related to research and artificial intelligence algorithms will be taught. Then, their implementation from different systems will be practiced. Students will practice the various algorithms on real problems.

Purpose: The primary purpose of the course is for students to understand and use basic concepts and techniques of artificial intelligence, as well as its application in concrete problems.

Expected results: After successful completion of this course, students will be able to: be equipped with basic knowledge of general algorithms on artificial intelligence; identify types of artificial intelligence

problems; formulate their own artificial intelligence problems; compare different versions of problems and apply them; evaluate and compare the performance of different artificial intelligence algorithms.

Forms of teaching and learning: Interactive learning, lectures, laboratory exercises, seminar papers, consultations.

Evaluation methods: Preparation and defense of project 1 (15%), Preparation and defense of project 2 (15%), Preparation and defense of project 3 (15%), Regular attendance (10%), Final exam (45%).

Grading

Points	Grade
0-49	5
50-60	6
61-70	7
71-80	8
81-90	9
91-100	10

Concretization tools: Laptops, smart screens, computers in laboratories, markers, pencils.

Relation between the theoretical and practical part of the study: 2:2.

Literature:

- Russell Stuart and Peter Norvig, Artificial intelligence: A modern approach, Prentice hall, Latest edition
- Sean Luke, Essentials of metaheuristics, Second Edition, 2012

Contribution on student load (must correspond with learning outcomes)			
Activity	Hours	Days/week	Total
Lectures	2	15	30
Exercise laboratory	2	15	30
Contact with lecturer/consultations	1	10	10
Mid-terms, seminars	2	3	6
Individual time spent studying (at the library or home)			75
Total			151

Subject: Theory of probability

ECTS: 6

Brief content: In this course, the basic concepts and statements from probability theory are given: elementary event space - sample space, probability and conditional probability, random variables, numerical parameters of random variables, most important discrete probability distributions and continuous, sum of independent random variables, limit theorems. These concepts are illustrated with practical examples from various scientific fields.

Purpose and expected results : The purpose of the course is to equip students with basic knowledge of the basis of the axiomatic construction of probability, the random variable and the meanings related to it, as well as limit theorems . After completing the course, the student will be able to:

- Explain and apply basic concepts from probability theory, including sample space, probability and conditional probability, independence of events;
- Describe and apply the concepts of discrete and continuous random variable;
- Formulate and apply properties of expectation and variance;
- Interpret and use basic distributions: binomial, Poisson, Pascal, gamma, beta and normal;
- I apply limit theorems, and in particular the central limit theorem;
- I apply the knowledge gained from this course to solve problems from various scientific fields and everyday life.

Forms of teaching and learning: Lectures, discussions, exercises, consultations, homework, colloquiums, exams.

Assessment methods: Homework (10%), First colloquium (15%), Second colloquium (15%), Final test (60%). The passing criterion is over 50% of the overall evaluation points. If the student does not pass the exam with colloquiums and homework, then the student is invited to the written exam weighing 40%, as well as the oral exam weighing 60%. Passing the written exam is necessary to take the oral exam.

Means of concretization: Marker, ordinary board, digital board, computer, projector.

The ratio between the theoretical part and exercises is 1:1

Basic literature

- B. Fejzullahu, *Basics of probability theory*, University of Pristina, Pristina, 2021.

Additional literature

- S. Ross, *A First Course in Probability*, 8th edition, Prentice Hall, New Jersey, 2009.
- W. Feller, *An Introduction to Probability Theory and its Applications I*, Wiley, New York, 1970.

Contribution to the student's workload (which must correspond to the student's learning outcomes)			
Activity	hour	Day/week	in total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30
Practical work			
Contacts with the teacher/consultations	1	15	15
Field exercises			
Colloquiums, seminars	2	2	4
Homework	2	15	30
Student's independent study time (in the library or at home)	2	15	30
Final exam preparation	5	1	5
Time spent on assessment (tests, quizzes, final exam)	2	3	6
Projects, presentations, etc.			
TOTAL			150

Subject: OOP and GUI Programming

ECTS: 6

Course content: Advanced Java interfaces are covered in this course, with special emphasis on event-driven programming for designing and coding applications, continuing with the Swing component needed for GUI applications.

Objectives of the course: Equipping students with additional knowledge of Java object-oriented programming and event-driven programming. Ability to implement advanced component structures and inheritance. Ability to implement GUI in applications. Upon successful completion of this course, students will be able to:

- Programming according to interfaces in Java;
- Implement advanced OOP programming
- Apply event-based programming to design and program Java applications;
- Implement the Swing component in Java to program GUI applications;
- Apply the knowledge gained from this course and connect the acquired products with other programming languages.

Teaching methodology: Lectures, laboratory exercises, group work, discussions, consultations.

Assessment methods and passing criteria: Periodic exam (25%), Laboratory work (15%), Regular follow-up (10%), Final exam (50%), Total (100%)

MARKS	Grading
< 51	5
51 to 60	6
61 to 70	7
71 to 80	8
81 to 90	9
> 90	10

Means of concretization: Marker, eraser, board, computer room, projector.

The ratio between the theoretical part and exercises The ratio between the theoretical and practical part is: 2 hours of theory and 2 hours of practice

Literature:

- D. Schmidt, *Programming principles in Java: architectures and interfaces* , Kansas State University, 2003.
- X. Jia, *Object oriented software development using Java* , Addison Wesley, 2000
- PS Wang, *Java with object-oriented programming* , Brooks /Cole-Thomson Learning, 2003.

Contribution to the student's workload (which must correspond to the student's learning outcomes)			
Activity	hour	Day/week	in total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30
Practical work	1.5	15	22.5
Contacts with the teacher/consultations	0.5	15	7.5
Field exercises			
Colloquiums, seminars	2	3	6
Homework	1	6	6
Student's independent study time (in the library or at home)	2	15	30
Final exam preparation			10
Time spent on assessment (tests, quizzes, final exam)			4
Projects, presentations, etc.			4

TOTAL			150 hours
--------------	--	--	------------------

Course title: Data security

ECTS: 6

Short description: In this course the topics include theoretical cryptography, symmetric cryptosystems, asymmetric key cryptosystems, one-way functions, digital signatures, secret schemes and the mathematical background needed to clarify them.

Course aim and expected results

They will learn the implementation of the theory of real and abstract data security systems, including recognition of potential threats of confidentiality and integrity. They will learn about the simple algorithms used since Roman times, symmetric DES and AES algorithms, asymmetric RSA algorithms, EL-Gamal, one-way functions SHA-1, SHA-2 and MD5 and will understand the diverse and multidisciplinary aspects in the field of data security. After completing the course, the student will be able to:

- Possess basic knowledge of data security,
- Be able to elaborate on strengths and weaknesses as well as the limitations of the main data security theories,
- To assess current trends in data security,
- Understand the historical development of data security and related issues,
- To analyze specific situations and to propose adequate solutions to ensure the security of computer systems,
- To explore the application of the theory of data security in the real world systems,
- To be able to understand the diverse and multidisciplinary aspects in the field of data security.

Teaching and learning forms: This course is explained through lectures, theoretical exercises, numerical exercises, homework and seminar work (where it is foreseen to be accomplished with continuous escort of assistant and lecturer).

Evaluation methods

Programming tasks: 15 point,
 Quizzes 5 point,
 Seminar: 10 point,
 First test 35 points,
 Second test 35 point.

Grading: Grading is done according to this scheme

Point	Grade
<50	5
>=50 and <60	6
>=60 and <70	7
>=70 and <80	8
>=80 and <90	9
>=90	10

Points from the programming tasks and seminar work that are achieved by the student during the course will be valid for each term. So the student is always subject to the exam with a maximum of 70 points.

Means of concretization / IT: Laptop, computer, projector, table.

The relationship between theoretical and practical part of the study 2:2

Literature:

- *Understanding Cryptography A Textbook for Students and Practitioners*, Christof Paar, Jan Pelzl 2010 Springer
- *Cryptography and Network Security - Principles and Practice 8th*, W. Stallings, Pearson, 2022

Contribution on student load (must correspond with learning outcomes)			
Activity	Hours	Days/week	Total
Lectures	2	15	30
Exercise laboratory	2	15	30
Practice work			
Contact with lecturer/consultations			
Field exercises			
Mid-terms, seminars	1	15	15
Homework	2	10	20
Individual time spent studying (at the library or home)	2	15	30
Final preparation for the exam	1	10	10
Time spent for final exam			
Quizzes	1	10	10
Projects, presentations, etc.	1	5	5
Total			150

Course: UX / UI Design

ECTS: 6

Course Description: This course provides an introduction to the principles and practices of User Interface (UI) and User Experience (UX) design. Students will learn about the importance of UI/UX in product design, how to conduct user research, create personas, design user flows and wireframes, and conduct usability testing. The course will cover both web and mobile design and students will gain practical skills using design software such as Sketch, Figma, and Adobe XD.

Cours aims: Students will gain a fundamental understanding of UI/UX design principles and practices, including the importance of user-centered design, conducting user research, creating personas, designing user flows and wireframes, and conducting usability testing.

Expected outcome: Upon completion of the course, students will be able to:

- Understand the role of UI/UX design in product development
- Conduct user research and create user personas
- Design user flows and wireframes for web and mobile applications
- Create high-fidelity designs using industry-standard design software
- Conduct usability testing and iterate designs based on feedback

Forms of Teaching: Lectures, Exercises, Quizzes, Laboratory Work, Discussions, Mid-term exams, Final exam

Grading Method: Attendance and class activity (10%), Semesterly practical project (30%), Lab work (15%), Final exam (45%)

Grades

piket	Nota
0-49	5
50-60	6
61—70	7
71-80	8
81-90	9
91-100	10

IT Concretization Tools: Cloud platforms such as AWS, Azure, or GCP, virtual machines, containers, orchestration tools, programming languages such as Python, and related software tools.

Report between theoretical and practical lecture hours: 2:2

Literature:

- Designing the User Interface: Strategies for Effective Human-Computer Interaction 6th Edition, Ben Shneiderman (Author), Catherine Plaisant (Author), Maxine Cohen (Author), Steven Jacobs (Author), Niklas Elmquist (Author), Nicholas Diakopoulos (Author)
- Don Norman (2013), The Design of Everyday Things, Basic Books
- Steve Krug (2014), Don't Make Me Think: A Common Sense Approach to Web Usability, New Riders
- Jesse James Garrett (2010), The Elements of User Experience: User-Centered Design for the Web and Beyond, Peachpit Press.

Contribution to the student's workload			
Activity	Hours	Day/Week	Total
Lectures	3	15	45
Theoretical/laboratory exercises	2	15	30
Contacts with the teacher/consultations	2	2	4
Colloquiums, seminars	2	2	4
Student's independent study time (in the library or at home)			67
Total load hours			150

Subject: Theory of matrices

ECTS : 6

Brief content : First, the basic meanings and notions related to matrices will be explained, then block matrices and block matrix operations will be examined. Next, some special types of matrices will be explained, and then elementary transformations by rows and columns will begin. In the following, some elementary concepts related to vector spaces will be explained, such as basis, space dimension, subspace, interspace operations, etc., continuing with the eigenvalues of the matrix, the determinant and the inverse matrix of the block matrix, the inverse of the sum, the rank of the product and the sum, and finally the eigenvalues of the matrix product.

Course goals : Deepening of knowledge in linear algebra and other areas of mathematics, with special emphasis on operator theory, statistics, operational research, computer science, engineering, etc.

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

- Perform block matrix operations;
- Bring the matrices into a suitable block matrix form;
- Find the eigenvalues of the matrix and the eigenvectors;
- Calculate the inverse of the sum for several types of matrices;
- Apply product eigenvalues in some special cases.

Teaching methodology : Lectures, theoretical exercises, group work, consultations.

Evaluation methods and passing criteria : Follow-up (10%), First test (20%), Second test (20%), Final exam (50%), Total (100%)

Means of concretization : Marker, pencil, board, computer, projector.

The ratio between the theoretical part and exercises is: 2:2

Literature:

- Charles G.Cullen, *Matrix and linear transformations* , Canada, 1990
- Zhang.F., *Matrix Theory* , Springer-Verlag New York, 1999
- R. Horn, C. Johnson, *Matrix analysis* , Cambridge University Press, 1999

Contribution to the student's workload (which must correspond to the student's learning outcomes)			
Activity	hour	Day/week	in total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30
Practical work			
Contacts with the teacher/consultations			
Field exercises			
Colloquiums, seminars	1	15	15
Homework	2	15	30
Student's independent study time (in the library or at home)	2	15	30
Final exam preparation	1	15	15
Time spent on assessment (tests, quizzes, final exam)			
Projects, presentations, etc.			
TOTAL	10	90	150

Course: Computing theory

ECTS: 6

Course Description: This course introduces the theoretical foundations of computer science, including formal language theory, automata theory, Turing machines, computability theory, and complexity theory. The course will cover the concepts of regular languages, context-free languages, and formal grammars. It will also cover the complexity of algorithms, including time and space complexity, NP completeness, and other topics. Students will learn about the relationship between theory and practice in computer science.

Course Goals: Provide the students with fundamental understanding of the theoretical foundations of computer science, including formal language theory, automata theory, Turing machines, computability theory, and complexity theory.

Expected Learning Outcomes: Upon completion of the course, students will be able to:

- Understand the basic concepts of formal language theory, automata theory, Turing machines, and computability theory
- Identify classes of problems that can be solved efficiently by a computer
- Analyse the time and space complexity of algorithms
- Understand the concept of NP completeness and other complexity classes
- Understand the relationship between theory and practice in computer science

Forms of Teaching: Lectures, Exercises, Quizzes, Laboratory Work, discussions, mid-term exams, final exam

Grading Method: attendance and class activity (10%), semesterly practical project (30%), lab work (15%), final exam (45%)

Grades

Piket	Nota
0-49	5
50-60	6
61—70	7
71-80	8
81-90	9
91-100	10

IT concretization tools: Lectures, problem sets, discussions, lab work, preliminary exams, final exam

Report between theoretical and practical lecture hours: 2 : 2

Literature:

- Michael Sipser (2006), Introduction to the Theory of Computation, Cengage Learning
- John Hopcroft and Jeffrey Ullman (1979), Introduction to Automata Theory, Languages, and Computation, Addison-Wesley
- Christos H. Papadimitriou (1993), Computational Complexity, Addison-Wesley

Contribution to the student's workload			
Activity	Hours	Day/Week	Total
Lectures	3	15	45
Theoretical/laboratory exercises	2	15	30
Contacts with the teacher/consultations	2	2	4
Colloquiums, seminars	2	2	4
Student's independent study time (in the library or at home)			67
Total load hours			150

Course: Linear programming

ECTS: 6

Short content: First we will explain the basic notions of linear programming course. Will be explained linear programming problem, the forms of linear programming, objective, terms of restrictiveness. Furthermore, will be studied geometric interpretation of linear programming including half spaces, hiperplanes, convex sets. Then wil be studied methods of solving linear programming problem with particular emphasis on simplex method. Then will be given meaning to the dual problem to the problem of linear programming data and links between them. Will be continued with the problem of linear programming with integer values. In the end will be discussed Transportation problem and methods for its solution.

Course objectives: Introduction of linear programming problem and the set of possible solutions. Geometric interpretation and solving linear programming problem with two variables. Solving of Linear programming problem by applying the simplex method. Introduction and formulation of the dual problem. Implementation of Gomory's method in finding solutions with integer values. Showing the transportation problem and methods for its solution.

Learning outcomes: After finishing the course students will be able to:

- Formulate the LPP.
- Solve the LPP with two variables using graphical method.
- Apply adequate method in problem solving.
- Formulate the dual problem from primal.
- apply Gomory's method in finding integer solutions,
- write the mathematical model of the given Transportation problem and solve it.

Teaching and learning methods: Lectures, discussions, exercises, homework, tests and final exam.

Method of evaluating outcomes and grading policy (assessment): Homework: 5%; regular attendance: 5%; first test 20%; second test 20%; (written exam 50%); final exam 50%; total 100%. Final grading: 50%-59% (six); 60%-69% (seven); 70%-79% (eight); 80%-89% (nine) and 90%-100% (ten).

Necessary and additional tools: Whiteboard, markers, eraser, computer and LCD projector.

Proportion between theory and exercise is 2:2.

Literature:

- Bernard Kolman, Robert E. Beck, Elementary linear programming with applications, 1995
- Louis Brickman, Mathematical Introduction to Linear Programming and Game theory, 1989

Student workload – Semester II			
Activity	Hours	Days/weeks	Total
Lectures	2	15	30
Exercise sessions (with TA)	2	15	30
Practical work	-	-	-
Office hours	1	15	15

Fieldwork	-	-	-
Midterms, seminars	2	2	4
Homework	2	15	30
Self-study	2	15	30
Final exam preparatin	5	1	5
Time spent in exams	2	3	6
Projects, presentations, etc	-	-	-
Total			150 hours

Semester V

Course: Programing for www

ECTS: 6

Short description: This course introduces students to web languages and technologies. It enables them to apply dynamic web programing techniques and how to successfully apply web programming concepts.

Goals and expected outcomes: The aim of this course is to equip students with the knowledge about Web technologies as well as web programing languages necessary for developing dynamic web sites. In this course students will learn to program in standard web languages: HTML/HTML5, CSS, JavaScript. Upon successful completion of this course, students should be able to:

- Design and implement simple or animated web sites using (X)HTML, CSS and JavaScript (DHTML).
- Design and implement interactive and dynamic web sites using (X)HTML, CSS and JavaScript (DHTML).
- Evaluate web sites for use on various browsers and platforms
- Know the standards and the role of W3C.

Teaching and learning methods: Lectures, discussions, hands-on exercises, consultations, homework, tests and exam.

Assesment: *Hands-on work: 30%, First test: 20%, Second test: 20%, Third test: 20%, Final exam: 10%.*

Points	Grading
0-49	5
50-61	6
62-73	7
74-85	8
86-97	9
98-100	10

Necessary and additional tools: Whiteboard markers, eraser, whiteboard, computer and LCD projector.

Proportion between theory and practice is 1:1

Basic Literature:

- Deitel & Andrew B. Goldberg, *Internet & World Wide Web How to program*, 2011
- Jon Duckett, *HTML and CSS: Design and Build Websites*, 2014
- Jon Duckett, *JavaScript and JQuery: Interactive Front-End Web Development*, 2014

Additional Literature:

- www.w3schools.com
- www.tutorialrepublic.com

Student Workload			
Activity	Hours	Days/weeks	Total
Lectures	2	15	30
Exercise sessions	2	15	30
Office hours	1	15	15
Midterms, seminars	1	6	6
Self-study	4	15	60
Projects, presentations, etc.	9	1	9
Total			150

Course: Algorithm analysis

ECTS: 6

Short content of the course: This course will enable students to use the mathematical tools to analyze the complexity of an algorithm. Students will be introduced to the symbolism of asymptotic analysis and some basic features and their implementation. Euclidean algorithm, binary search algorithm, insertion and sorting algorithms, will also be studied. Students will also be introduced with elements from Dynamic Programming and the full NP class problems.

Course objective: The main goal of the course will be to enable students to designate an algorithm to consider its complexity, which is very important in the field of programming. This will enable students from some algorithms to solve a given problem, to choose the algorithm whose complexity is the lowest, preferably polynomial.

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

- Understand the importance of the complexity of an algorithm;
- Easily manipulate the symbols of asymptotic analysis;
- use some of the dynamic programming techniques to solve some problems in the field of computer science;
- Define the class of NP-s problems and full NP classes and provide some examples of these problems.

Teaching and learning methods: Lectures, discussions, exercises, consultations, projects, colloquia, exams.

Assessment : Attendance 10%; Periodic Exam 15%; Periodic Exam II: 15%; seminar paper 20%, final exam 40%.

Necessary and additional tools: Whiteboard markers, eraser, whiteboard, computer and LCD projector.

Proportion between theory and exercise is 2:2.

Literature

- Clifford A. Shaffer, Data Structures and Algorithm Analysis, USA 2011
- E. Bajrami, Lecture slides and Exercise scripts
- Sara Baase: *Computer Algorithms-Introduction to Design and Analysis*, Addison-Wesley Publishing Company, 1989.
- Thomas H. Cormen; Charles E. Leiserson; Ronald L. Rivest: *Introduction to Algorithms* MIT Press, McGraw-Hill Book Company, 1999.

Contribution on student load			
Activity	Hours	Day/ Week	Total
Lectures	2	15	30
Exercise sessions (with TA)	2	15	30
Contacts with the teacher / consultants	1	15	15
Midterms, seminars			4
Student study time (in library or at home)	3	15	45
Final preparation for exam			15
Eveluation time			4
Project presentation, etc.	3	4	12
Total			155

Course: Statistics

ECTS: 6

Short description: Basic concepts of statistics. Presentation of statistical data. Numerical characteristics of center of empirical distribution: arithmetic mean, median, mode, geometric mean, harmonic mean. Measures of Spread: Quartiles and Interquartile Range, average absolute deviation, standard deviation. Moments, asymmetry, absolute measure of asymmetry, relative measure of asymetry. Simple Random Sampling from “Random Numbers”. Empirical mean and dispersion of smples, frequency's distribution of sample. Gama function, χ^2 distribution, student’s distribution $t(n)$. Testing a Statistical Hypothesis, basic concepts. Types of error. Parametric hypothesis testing for μ . Parametric hypothesis testing for σ^2 . Non-parametric statistical hypothesis. Non-parametric statistical hypothesis. Applications. Implementations in the R language.

Objective: The purpose of the course is to provide students with data processing and analysis, graphical data display and basic methods and statistical analysis procedures required for independent operations, as well as statistical analysis of cases from real business practice.

Expected results: Upon completion of this course students will be able to:

- Analyze basic concepts and definitions of probability and statistics,
- Analyze the results of the research,
- Understand the results of data analysis,

- Recognize the need to use statistical data processing,
- Demonstrate their knowledge of the basics of inferential statistics by making valid generalizations from sample data,
- Use R and Python to perform statistical analysis.

Methodology of teaching: Lectures, discussions, exercises, laboratory exercises, consultations, homework, colloquiums, exams.

Methods of assessment: First test (10%), Second test (10%), Seminar (20%), Final test (60%).

Points	Mark
< 51	5
51 - 60	6
61 - 70	7
71 - 80	8
81 - 90	9
> 90	10

Concretization tools: Marker, board, computer room, projector, laptop.

Ration between Theoretical part and exercises: 2:2

References

1. Sh. Leka, *Teoria e Probabiliteteve dhe Statistika Matematike*, Tiranë, 1998.
2. J. Gareth, D. Witten, T. Hastie & R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*, New York, 2013: Springer.
3. R. Kabacoff, *R In Action: Data Analysis and Graphics with R*, Shelter Island, New York, 2015: Manning Publications Co.

Student Workload			
Activity	Hours	Days/weeks	Total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30
Contacts with the teacher/consultations	1	15	15
Midterms, seminars			25
Student's independent study time (in the library or at home)			50
Total load hours			150

Course title: Distributed programming

ECTS: 6

Short description: This course introduces students to distributed systems and programming for the same. Students are introduced to challenges of programming for distributed systems and the application of various techniques in order to overcome these challenges.

Course aim: The aim of this course is for students to become familiar with the basic concepts of computer communication, processes and threads, Sockets, Remote Method Invocation, and Multicast Sockets.

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

- Understand and explain the reasons behind fundamental and potential problems in distributed systems.
- Explain and describe various distributed system models, including error systems and timing models.
- Understand the role of Client-Server operations
- Use sockets in order to implement Client-Servers communication programs
- Understand remote method invocation

Teaching and learning forms: Lectures, discussions, exercises, consultations, homework, colloquiums and exams.

Evaluation methods:

Programming tasks: 15 point,
 Quizzes 5 point,
 Seminar: 10 point,
 First test 35 point,
 Second test 35 point.

Grading: Grading is done according to this scheme

Point	Grade
<50	5
>=50 and <60	6
>=60 and <70	7
>-70 and <80	8
>=80 and <90	9
>=90	10

Points from programming tasks and seminar work that are achieved by the student during the course will be valid for each term. So the student is always subject to the exam with a maximum of 70 points.

Means of concretization / IT: Laptop, computer, projector, table.

The relationship between theoretical and practical part of the study 2:2

Literature:

- George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair: *Distributed Systems: Concepts and Design, 5th Edition, 2011*
- Raja Malleswara, Rao Pattamsetti, *Distributed Computing in Java 9*, Packt, 2017
- Y. Daniel Liang, *Introduction to Java Programming, Comprehensive 12/E*, Pearson

Contribution on student load (must correspond with learning outcomes)			
Activity	Hours	Days/week	Total
Lectures	2	15	30
Exercise laboratory	2	15	30
Practice work			
Contact with lecturer/consultations	1	15	15
Field exercises			
Mid-terms, seminars	2	10	20
Homework			

Individual time spent studying (at the library or home)	2	15	30
Final preparation for the exam	1	10	10
Quizzes	1	10	10
Projects, presentations, etc.	1	5	5
Total			150

Course: Accounting

ECTS : 6

Short content:

Accounting elaborates, both theoretically and practically:

- General knowledge of the accounting system and procedures for creating accounting information,
- Transaction registration forms,
- The application of double-entry accounting,
- Elements of financial statements,
- Completing the accounting cycle,
- Recognition of income, expenses,
- The accounting of commercial companies, the state of goods and their registration,
- Accounting of fixed and intangible assets, etc.

Goals:

- Gaining new theoretical and practical knowledge from the field of accounting.
- Providing real opportunities for the creation of a professional framework for the application of acquired accounting knowledge, for the recognition, processing, recording of financial transactions in business entities as well as
- Creation of financial statements and accompanying documents.

Expected results: After completing this course, students will be able to:

- Create accounting information for making business decisions,
- Recognize and record financial transactions as well as the compilation of financial statements and accompanying documents,
- Recognize and account for the expenses, income and financial result of the entities, depending on the level of the business relying on the International Accounting Standards (IAS) and the Kosovo Accounting Standards (SKK),
- Deepen the knowledge about the accounting tracking of assets and sources of assets,
- Deepen the knowledge about the way of financing the entity, maintaining the substance of the entity and reporting changes in equity, etc.

Forms of teaching and learning : Teaching will take place through lectures, practical tasks, individual and group interpretations, periodic self-assessments, etc. All these will be realized in the theoretical and practical aspect by presenting the materials in audio-visual form through electronic technology with Windows Office programs. In the theoretical aspect, general scientific knowledge will be offered, based on contemporary literature. The practical part will mainly be realized through concrete examples from the literature and from practical knowledge of private and public production enterprises and profit-

making organizations. In this way, the creation of interactive professor-student and student-student relationships will be aimed.

Assessment methods and passing criteria: Student assessment will be based on:

- of their continuity and commitment,
- written assignments
- success in colloquia as well
- the final exam.

Means of concretization: Use of the Microsoft Office program, projector and laptops as well as computers in the computer room for each student, when solving tasks in the classroom.

The ratio between the theoretical part and exercises is 2:2

Basic literature:

1. Prof. dr. Skender Ahmeti: "Financial Accounting, (basic concepts)" Prishtina, 2012
2. Prof. dr. Rustem Asllanaj: "Financial accounting", Prishtina, 2013

Additional literature :

1. Robert Ingram, Thomas Albright and Bruce Baldwin: "Financial Accounting" Fifth Edition, Thompson Learning, Cincinnati, Ohio, 2003,
2. King, Lemble & Smith: "Financial accounting - A decision making approach" Second Edition, John Wiley and Sons, New York, 2001
3. John Wild: "Financial accounting - Information for decision" Third Edition, McGraw - Hill, 2005

Rating:

- Activity during the semester (5%)
- colloquiums (80%)
- Final exam (15%)

Marks

≥45 enough (6)

≥60 good (7)

≥70 very good (8)

≥80 excellent (9)

≥90 distinguished (10)

Students who do not meet the requirements to pass the course, then they must retake the course in the next academic year.

Contribution to the student's workload			
Activity	hour	Day/week	in total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30

Practical work			
Contacts with the teacher/consultations	2	15	30
Field exercises			
Colloquiums, seminars	2	2	4
Homework			
Student's independent study time (in the library or at home)	2	15	30
Final exam preparation	22	1	22
Time spent on assessment (tests, quizzes, final exam)	2	2	4
Projects, presentations, etc.			
TOTAL			150 hours

Course: Internet of Things (IoT)

ECTS: 6

Course Description: Fundamentals of IoT- Simplified IoT Architecture and Core IoT Functional Stack – Fog, Edge and Cloud in IoT; IoT Protocols - Physical and MAC layers, topology and Security of IEEE 802.15.4, 802.15.4g, 802.15.4e, 1901.2a, 802.11ah and LoRaWAN – Network Layer: IP versions; Design and Development - Design Methodology – Embedded computing logic – Microcontroller, System on Chips – IoT system building blocks; Data Analytics and Supporting Services; Case Studies/Industrial Applications - IBM Watson IoT platform

General Competencies: Students will gain fundamental knowledge and understanding of Smart Objects and IoT Architectures, will learn about various IOT-related protocols, will build simple IoT Systems using Arduino and Raspberry Pi, and will also understand data analytics and cloud in the context of IoT. They should develop IoT infrastructure for popular applications.

Learning Outcomes: Upon completion of this course, students will be able to:

- Explain the concept of IoT.
- Analyze various protocols for IoT.
- Design a PoC of an IoT system using Raspberry Pi/Arduino
- Apply data analytics and use cloud offerings related to IoT.
- Analyze applications of IoT in real time scenario

Forms of Teaching: Lectures, Exercises, Quizzes, Laboratory Work, discussions, mid-term exams, final exam

Grading Method: attendance and class activity (10%), semesterly practical project related to IoT applications (40%), quizzes (5%), final exam (45%)

Grades

piket	Nota
0-49	5
50-60	6
61—70	7
71-80	8

81-90	9
91-100	10

IT concretization tools: laptop, projector, lab computers, marker, eraser, Rasperry Pi/Arduino microcontrollers

Report between theoretical and practical lecture hours: 2 : 2

Literature:

- David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, —IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, Cisco Press, 2017
- Jan Höller, Vlasios Tsiatsis, Catherine Mulligan, Stamatis, Karnouskos, Stefan Avesand. David Boyle, “From Machine-to-Machine to the Internet of Things – Introduction to a New Age of Intelligence”, Elsevier, 2014.

Contribution to the student's workload			
Activity	hour	Day/week	in total
Ligjërata	2	15	30
Ushtrime teorike/laboratorike	2	15	30
Kontaktet me mësimdhënësin/konsultimet	1	15	15
Kolokiume, seminare	2	10	20
Koha e studimit vetanak të studentit (në bibliotekë ose në shtëpi)	2	15	30
Përgatitje për provim	1	10	10
Teste, kuize	1	10	10
Projekte, prezantime	1	5	5
Ngarkesa totale orë			150

Course title: Game development

ECTS: 6

Short description: This is a course oriented towards projects on designed game design and game programming project. Students will work in teams to design, implement, and test interactive, two- and three-dimensional games with animation, voice, limitations, and networking capabilities. Software engineering aspects and aspects of the development of a video game will be covered.

Course aim and expected results: Equipping students with basic knowledge of software development. Training for the implementation of computer graphics for visualization. Training for game programming with two- and three-dimensional graphics. After completing the course, the student will be able to:

- understand the principles of software development;
- learn about two-dimensional computer graphics;
- learn about three-dimensional computer graphics;
- implement sprawling manipulation, scroll and other playground techniques;
- design software games;
- Implement computer software games.

Teaching and learning forms: Lectures, exercises, quizzes during exercises, seminary work, classroom discussions, consultations.

Evaluation methods:

Programming tasks: 15 point,
 Quizzes 5 point,
 Seminar: 10 point,
 First test 35 point,
 Second test 35 point.

Grading: Grading is done according to this scheme

Point	Grade
<50	5
>=50 and <60	6
>=60 and <70	7
>=70 and <80	8
>=80 and <90	9
>=90	10

Points from programming tasks and seminar work that are achieved by the student during the course will be valid for each term. So the student is always subject to the exam with a maximum of 70 points.

Means of concretization / IT: Table, computer, projector.

The relationship between theoretical and practical part of the study 2:2

Literature:

- *Java Game Development with LibGDX from Beginner to Professional*, Lee Stemkoski, 2018, Apress
- *Introduction to Java Through Game Development*, Victor Brusca, 2022, Apress

Contribution on student load (must correspond with learning outcomes)			
Activity	Hours	Days/week	Total
Lectures	2	15	30
Exercise laboratory	2	15	30
Practice work			
Contact with lecturer/consultations	1	15	15
Field exercises			
Mid-terms, seminars	2	10	20
Homework			
Individual time spent studying (at the library or home)	2	15	30
Final preparation for the exam	1	10	10
Quizzes	1	10	10
Projects, presentations, etc.	1	5	5
Total			150

Course: Game Theory

ECTS: 6

Short description: This course enables students to study and use notions and techniques from Game Theory. We will give many examples from real-life and basic ideas of games of strategy. We will study games with sequential moves and with simultaneous moves, and their combination. We will also treat zero-sum games and non-zero sum games, as well as uncertainty and information in a game. Finally, strategic moves, prisoners' dilemma, repeated games, collective-action games, and evolutionary games will also be studied.

Goals: This course aims to help students to understand and use the fundamental notions and techniques of game theory, and to apply them in concrete financial as well as non-financial problems.

Expected outcomes At the end of the course students should be able to:

- Identify, explain, use and analyze games of strategy.
- Analyze some basic properties of games of strategy.
- Understand the difference between decisions and games.
- Identify problems from various fields and use mathematical models from game theory to solve them.
- Apply the knowledge gained in fields outside mathematics.
- Create models of games for application outside mathematics.

Teaching and learning methods: Lectures, discussions, exercises, consultations, homework, tests and exam.

Assessment: Participation (10%); Midterms (2 x 15%); Homework (15%); Projects (15%); Final exam (30%).

Grading schema:

Points	Grade
0-49	5
50-60	6
61-70	7
71-80	8
81-90	9
91-100	10

Necessary and additional tools: Whiteboard markers, eraser, whiteboard, computer and LCD projector.

Proportion between theory and practice is 2:2

Literature: Open Yale Courses Econ 159: <https://oyc.yale.edu/economics/econ-159>

A. Dixit, S. Skeath & D. Reiley, [*Games of Strategy \(3rd Edition\)*](#), W.W. Norton & Co. 2009, New York.

Additional literature:

Peter Morris, *Introduction to Game Theory*, Springer-Verlag, 1994 New York.

[Martin J. Osborne](#) & [Ariel Rubinstein](#), *A Course in Game Theory*, 1994 MIT Press.

Student Workload			
Activity	Hours	Days/weeks	Total
Lectures	2	15	30

Exercise sessions	2	15	30
Office hours	1	15	15
Midterms, seminars	2	2	4
Self-study	3	15	45
Projects, presentations, etc.	3	4	12
Time spent in exams	4		4
Final exam preparation	10		10
Total			150

Course : Information systems

ECTS : 6

Course content: The course will introduce students to the concepts of information systems in management and their applications. During the lectures and exercises, the connection between information and business, business benefits from the use of information technology, decision-making based on data, development of systems, protection of systems, people and information, future trends will be elaborated .

Objectives of the course: inform students about the role of information systems and their use by businesses; enable them to gain theoretical knowledge of information systems; understand the importance of information systems in contemporary businesses.

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

- Understand the fundamental concepts of information systems
- Analyze the role of information systems in organizational strategy and competitive advantage:
- Design and implement information systems solutions
- Evaluate the ethical and social implications of information systems
- Evaluate the impact of emerging technologies on information systems
- Communicate effectively about information systems

Forms of teaching and learning: Lectures, discussions, exercises, consultations, homework, tests and exams.

Assessment methods: Practical work: 30%, First test: 20%, Second test: 20%, Third test: 20%, Final exam: 10%.

Grading scheme:

Points	Grading
0-49	5
50-61	6
62-73	7
74-85	8
86-97	9
98-100	10

Means of concretization: digital board, computer, projector.

The ratio between the theoretical and practical part is: 1:1

LITERATURE

- Brad Prince, R. Kelly Rainer, : *Introduction to Information Systems, 9th Edition, 2021*
- Stephen Haag, Maeve Cummings, *Management Information Systems for the Information Age, 9th edition, McGraw-Hill-Irwin, 2012*

Contribution to the student's workload			
Activity	Hour	Day/week	in total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30
Contacts with the teacher/consultations	1	15	15
Colloquiums, seminars	1	6	6
Student's independent study time (in the library or at home)	4	15	60
Projects, presentations	9	1	9
Total load hours			150

Semester VI

Subject: Project (industrial practice)

ECTS: 6

Description of the subject: Students will apply for internships in companies with which the Department makes agreements. Also, they will select the Professor from the Computer Science courses, who will supervise them and grade them.

Purpose: Students to finish practical professional work in a local company.

Expected results: To introduce students to the industry and the correlation of acquired skills with business requirements; enable them to gain experience in a development company in Kosovo; understand the importance of working in larger groups and working in industry in general.

Forms of teaching and learning: Visit to the company, division of tasks in cooperation with the industry, presentation of completed work, consultations.

Evaluation methods: Presentation of the idea (10%), First presentation of the project (30%), Final presentation of the project (60%). If the student does not pass the course, then he must take it again the next year.

Grading

Points	Grade
0-49	5
50-60	6
61-70	7
71-80	8
81-90	9
91-100	10

Concretization tools: Laptops, smart screens.

Relation between the theoretical and practical part of the study: 0:4;

Literature:

In accordance with the lecturer.

Contribution on student load (must correspond with learning outcomes)			
Activity	Hours	Days/week	Total
Lectures			
Exercise laboratory			
Contact with lecturer/consultations			
Mid-terms, seminars	1	3	3
Individual time spent studying (at the library or home)			150
Total			153

Course title: Image Processing and Computer Graphics

ECTS: 6

Short description: This course covers the core part of Image Processing and Computer Graphics. These as two separate disciplines but interrelated, play an increasingly important role in contemporary applied Computer Science. In addition to the basic concepts, also the concepts of modeling will be taught: primitive objects, curves, surfaces, scene graphs, lighting, shadow, textures. Then we give the definition and representation of the image, color patterns, gray level, color correction, neighborhood-based operations, filtering, noise removal will be given. Image processing will be learned through MatLab while Computer Graphics will be learned through the Java programming language.

Course aim: The goals of this course are the basic understanding of image processing and computer graphics. Application of image processing in science, application of elemental geometry in image processing, image filtering in the spatial domain, use of various filtering operators, and the importance of mathematics in building these operators. Application of computer graphics in science, algorithms for drawing two-dimensional primitive objects, creation of two-dimensional and three-dimensional objects as well as transformations with them, programming in computer graphics.

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

- Have a complete overview of image processing and computer graphics
- To read, write and process images (using MatLab)
- Apply affine transformations to images and objects
- Have the opportunity to develop programming skills in image processing and computer graphics through programming tasks.
- Parametric curves as well as their graphical layout
- Have basic knowledge in light and shadow modeling

Teaching and learning forms: This course is explained through lectures, theoretical exercises, assignments and seminar work (where it is foreseen to be accomplished with constant escort of assistant and lecturer of the subject).

Evaluation methods

Programming tasks: 15 point
Quizzes 5 point,
Seminar: 10 point
First test 35 points,

Second test 35 points.

Grading: Grading is done according to this scheme

Point	Grade
<50	5
>=50 and <60	6
>=60 and <70	7
>-70 and <80	8
>=80 and <90	9
>=90	10

Points gained from programming work and seminar during the course will be valid for each term. So the student always undergoes the exam with a maximum of 70 points.

Means of concretization / IT: Laptop, projector, computers in laboratories, table, markers, sponge.

The relationship between theoretical and practical part of the study: 2:2

Literature:

- *Digital Image Processing using Matlab*, Rafael C. Gonzales, University of Tennessee
- Richard E. Woods, MedData Interactive, Steven L. Eddins, The MathWorks, Inc, Prentice Hall, 2004,
- *Computer Graphics Using Java™ 2D and 3D*, Hong Zhang, Y. Daniel Liang – Armstrong Atlantic State University, Prentice Hall, December 06,2006,
- *Computer Graphics for Java Programmers, Second Edition*, Leen Ammeraaland Kang Zhang, John Wiley & Sons, 2007.

Contribution on student load (must correspond with learning outcomes)			
Activity	Hours	Days/week	Total
Lectures	2	15	30
Exercise laboratory	2	15	30
Practice work			
Contact with lecturer/consultations	1	15	15
Field exercises			
Mid-terms, seminars	2	10	20
Homework			
Individual time spent studying (at the library or home)	2	15	30
Final preparation for the exam	1	10	10
Quizzes	1	10	10
Projects, presentations, etc.	1	5	5
Total			150

Course: Software engineering

ECTS: 6

Course content: In this course, the understanding of the problems involved in the development of high-quality software, as well as the techniques and tools necessary to develop such systems, will be elaborated. The subject mainly focuses on the development processes of software projects (traditional and contemporary ones), system modeling and therefore its documentation through different stages, as well as software project management.

Objectives of the course: The main objectives are the presentation and discussion of traditional and contemporary software development processes, systems testing practices, modern software project management practices including time, risk, and system complexity assessment.

Learning outcomes: After completing this course students should be able to:

- Clarify the need and importance of Software Engineering;
- Propose adequate processes for software development;
- Evaluate the advantages and disadvantages of different software processes;
- Model different parts of the system, using the "UML" modeling language;
- Evaluate the "size" of software projects, applying different models;
- Plan the time needed for software development activities.

Forms of teaching and learning: Lectures, discussions, exercises, consultations, homework, tests and exams.

Assessment methods: Practical work: 30%, First test: 20%, Second test: 20%, Third test: 20%, Final exam: 10%.

Grading scheme:

Points	Grading
0-49	5
50-61	6
62-73	7
74-85	8
86-97	9
98-100	10

Means of concretization: digital board, computer, projector.

The ratio between the theoretical and practical part is: 1:1

Literature:

- Frank Tsui, Orlando Karam, et al.: *Essentials of Software Engineering 5th ed.* , 20 22 .
- Ian Sommerville, *Software Engineering, 10th^{edition}* , Addison-Wesley, 2018 .

Contribution to the student's workload			
Activity	hour	Day/week	in total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30
Contacts with the teacher/consultations	1	15	15
Colloquiums, seminars	1	6	6
Student's independent study time (in the library or at home)	4	15	60
Projects, presentations	9	1	9
Total load hours			150

Subject: Data analysis

ECTS: 6

Short content: This course includes an introduction to data analysis and its techniques. The course contains five main parts, starting with data preprocessing, which takes up most of the time during data

analysis, then continues with visualization, predictive models, analysis of relationships between data, data clustering and anomaly detection.

Purpose: Familiarizing students with the basic concepts and algorithms assigned to each major part, thus giving them the opportunity to apply the models to real problems.

Expected results: Upon successful completion of this course, students will be able to: manipulate data used for analysis; understand predictive models and their algorithms; find connections between data and use their algorithms; group data based on common characteristics; detect the occurrence of anomalies and finally visualize the results of all the mentioned techniques.

Forms of teaching and learning: Interactive learning, lectures, laboratory exercises, seminar papers, consultations.

Evaluation methods: Seminar Paper 1 (15%), Seminar Paper 2 (15%), Seminar Paper 3 (15%), Seminar Paper 4 (15%), Periodic Exams (40%).

Grading

Points	Grade
0-49	5
50-60	6
61-70	7
71-80	8
81-90	9
91-100	10

Means of concretization / IT: Laptops, smart screens, computers in laboratories, markers, pencils.

The ratio between the theoretical and practical part of the study 2:2

Literature:

- P. Tan, M. Steinbach, A. Karpatne, Y. Kumar, Introduction to data mining, Pearson, Latest Edition
- Jiawei Han, Micheline Kamber, Jian Pei, Data Mining: Concepts and Techniques, The Morgan Kaufmann Series in Data Management Systems, Latest Edition

Contribution on student load (must correspond with learning outcomes)			
Activity	Hours	Days/week	Total
Lectures	2	15	30
Exercise laboratory	2	15	30
Contact with lecturer/consultations	1	10	10
Mid-terms, seminars	2	3	6
Individual time spent studying (at the library or home)			75
Total			151

Course : Programming in www server

ECTS : 6

Course content: An intermediate-level examination of the theory and practice of developing server-side applications for www. Students will learn practical techniques for designing and implementing data-driven web pages through the use of server-side processing.

Objectives of the course: As a continuation of the www Programming course, in addition to creating live web sites and increasing their programming skills, students will learn how to integrate PHP language into HTML, database interaction, access issues, transfer of files, file management and server configuration (Apache).

Learning outcomes: After completing this course students should be able to:

- Describe the issues and methodologies involved in developing data-driven websites
- Analyze web development problems based on those facts
- Apply those methodologies to develop appropriate solutions based on analysis, and
- Evaluate the efficiency of those solutions
- Develop proficiency in web server administration
- Apply best practices for web server programming

Forms of teaching and learning: Lectures, discussions, exercises, consultations, homework, tests and exams.

Assessment methods: Practical work: 30%, First test: 20%, Second test: 20%, Third test: 20%, Final exam: 10%.

Grading scheme:

Points	Grading
0-49	5
50-61	6
62-73	7
74-85	8
86-97	9
98-100	10

Means of concretization: digital board, computer, projector.

The ratio between the theoretical and practical part is: 1:1

LITERATURE:

- Jon Duckett: *PHP & MySQL: Server-side Web Development, 2019*
- Deitel & Andrew B. Goldberg, *Internet & World Wide Web How to program, 2011*
- Tom Butler and Kevin Yank: *PHP & MySQL Novice to Ninja 6th Edition, 2017*

Contribution to the student's workload			
Activity	Hour	Day/week	Total
Lectures	2	15	30
Theoretical/laboratory exercises	2	15	30
Contacts with the teacher/consultations	1	15	15
Colloquiums, seminars	1	6	6
Student's independent study time (in the library or at home)	4	15	60
Projects, presentations	9	1	9
Total load hours			150

Course: Software development in OOP

ECTS: 6

Course Description: The course will introduce the following concepts: The basic concepts of object-oriented programming. Classes and object. Programming in Java. Static class members. Inheritance. Polymorphism. Abstract classes and interfaces. Exception handling. Working with files in Java. Windows programming in Java.

General Competencies: The course will introduce concepts and methodology of object-oriented programming in Java. Goal of the course is to teach participants how to abstract a problem in an object-oriented style and program a solution in Java for that problem. Students will also learn to develop Windows applications by using Java IDE (Integrated Development Environment). Special attention will be given to practical work.

Learning Outcomes: Mastering the object-oriented programming and skills to independently design solutions to problems using the object-oriented concept and to learn how to develop applications in Java programming language. Students will acquire:

- strong understanding of OOP concepts,
- good understanding of standard Java libraries,
- further develop real-life problem-solving skills.
- Developing maintainable and scalable code
- Testing and debugging software
- Collaborating on software development projects

Forms of Teaching: Lectures, Exercises, Quizzes, Laboratory Work, discussions, mid-term exams, final exam

Grading Method: attendance and class activity (10%), semesterly practical project (30%), lab work (15%), final exam (45%)

Grades

Piket	Nota
0-49	5
50-60	6
61—70	7
71-80	8
81-90	9
91-100	10

IT concretization tools: laptop, projector, lab computers, marker, eraser.

Report between theoretical and practical lecture hours: 2 : 2

Literature:

- Introduction to Java Programming, Comprehensive Version, 11th Edition- Y. Daniel Liang
- X. Jia, Object oriented software development using Java: principles, patterns and frameworks, Addison Wesley, 2002.

Contribution to the student's workload			
Activity	Hour	Day/week	Total
Ligjërata	2	15	30
Ushtrime teorike/laboratorike	2	15	30
Kontaktet me mësimdhënësin/konsultimet	1	15	15
Kolokiume, seminare	2	10	20
Koha e studimit vetanak të studentit (në bibliotekë ose në shtëpi)	2	15	30
Përgatitje për provim	1	10	10
Teste, kuize	1	10	10
Projekte, prezantime	1	5	5
Ngarkesa totale orë			150

Course title: Parallel programming

ECTS: 6

Short description: Parallel programming methods, building parallel algorithms from sequential ones and applying parallel programming to those algorithms, in order to shorten the computation time.

Course aim and expected results: With parallel programming students will learn programming in multi-core processors, programming in the graphics card processor (GPU). They will learn to program and solve problems (algorithms) in Cryptography, Imaging Processing, Computer Graphics, Mathematics, etc. After completing the course, the student will be able to:

- Have a complete overview of parallel programming,
- Implement applications using Java, C ++,
- Describe different parallel architectures, programming models, and common algorithms such as multiplication of vector with matrix,
- Understand the problems provided by finding a parallel algorithm,
- Analyzing the complexity of that algorithm in function of the number of processors,
- Implementing algorithms using Message Passing Interface (MPI) and CUDA (Computed Unified Device Architecture)
- Implementation of sequential algorithms in the parallel ones.

Teaching and learning forms: This course is explained through lectures, theoretical exercises, homework assignments and seminar work (which is foreseen to be carried out with constant follow up of the assistant and lecturer of the subject).

Evaluation methods

Programming tasks: 15 point,
 Quizzes 5 point,
 Seminar: 10 point,
 First test 35 point,
 Second test 35 point.

Grading: Grading is done according to this scheme

Point	Grade
<50	5
>=50 and <60	6
>=60 and <70	7
>-70 and <80	8

≥ 80 and < 90	9
≥ 90	10

Points from programming tasks and seminar work that are achieved by the student during the course will be valid for each term. So the student is always subject to the exam with a maximum of 70 points.

Means of concretization / IT: Laptop, computer, projector, table.

The relationship between theoretical and practical part of the study 2:2

Literature:

1. *Parallel and Distributed Programming Using C++*, Cameron Hughes and Tracey Huhges, Addison Eesley, 2004.
2. *Programming Massively Parallel Processors – A Hands on Approach*, David B. Kirk and Wen-mei W. Hwu, Morgan Kaufman, 2010.

Contribution on student load (must correspond with learning outcomes)			
Activity	Hours	Days/week	Total
Lectures	2	15	30
Exercise laboratory	2	15	30
Practice work			
Contact with lecturer/consultations	1	15	15
Field exercises			
Mid-terms, seminars	2	10	20
Homework			
Individual time spent studying (at the library or home)	2	15	30
Final preparation for the exam	1	10	10
Quizzes	1	10	10
Projects, presentations, etc.	1	5	5
Total			150

Course: Stochastic Processes

ECTS: 6

Short content of the course: This course examines the conditional phenomena, such as free movement model, Markov chains in discrete and continuous time, Poisson's processes, birth and death, renewal and reward processes etc.

Course objective: The aim of the course is to familiarize students with the basic meanings of stochastic processes such as free movement, discrete and continuous processes, Markov processes, Poisson processes and stationary processes.

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

- Identify and use random processes,
- apply in the field of business and computer science,
- Apply Markov processes in concrete cases,
- Analyze natural phenomena in terms of stochastic processes as well
- Distinct discrete and continuous processes

Teaching and learning methods: Lectures, discussions, exercises, consultations, projects and exams.

Assessment: Attendance 10%; Periodic Exam 15%; Periodic Exam II: 15%; Projects 20%, final exam 40%.

Necessary and additional tools: Whiteboard markers, eraser, whiteboard, computer and LCD projector.

Proportion between theory and exercise is 2:2.

Literature:

- Sheldon M. Ross, Introduction to Probability Models, 10th Ed.;Elsevier 2010.
- E. Bajrami, Lecture slides-notes
- Saeed Ghahramani, Fundamentals of probability with stochastic processes, 3th Ed.; Boston, London: Addison-Wesley, 2005.

Contribution on student load			
Activity	Hours	Day/ Week	Total
Lectures	2	15	30
Exercise sessions (with TA)	2	15	30
Contacts with the teacher / consultants	1	15	15
Midterms, seminars			4
Student study time (in library or at home)	3	15	45
Final preparation for exam			10
Evaluation time			4
Project presentation, etc.	3	4	12
Total			150