UNIVERSITY OF TIRANA
FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

D E A N
Prof. Dr. Spiro DRUSHKU

FIRST CYCLE OF STUDY
BACHELOR IN INFORMATICS
BRANCH PROGRAM
2018 – 2021

HEAD OF DEPARTMENT
Prof. Asoc. Endrit XHINA

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

First Year Courses

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Introduction to computer science

Teaching Activities	Lectures	Exercises	Lab. Work	Practice	Total
Frequentation Requirement	Non mandatory	60%	80%	0%	0
Teaching Hours	30	60	15	0	105
Teaching Hours	170				
Language of Lectures	Albanian				
ECTS Credits	11				
Lectures	First Year, First Semester, 15 weeks: 2 lecture hours, 4 exercise hours, 1 lab hour				
Evaluation	Written exam and Laboratory work				
Grading	Laboratory programs			30%	
	Preparation and Participation			5%	
	Final Exam			65%	
	Total			**100%**	
Instructor	**MSc. Brisilda MUNGULI**				

Course Overview	This first course in computer science develops foundational skills in computer programming using pseudocode and flowcharting. The course will introduce the process of developing algorithms to solve problems, and the corresponding process of developing computer programs to express those algorithms.
Objectives	Providing basic computational thinking knowledge for problem solving.
Preliminary Requirements	Students should know the basics of using a computer and should be comfortable with math.

Lecture Topics

1. Introduction
Computer System. Information representation in binary system. **(2 hours)**

2. Binary Arithmetic
Adding, multiplying, subtracting and dividing binary numbers. **(2 hours)**

3. Functional Components of a Computer System
Input Unit, Output Unit, CPU and Memory Unit. **(2 hours)**

4. Information Encoding. Main Computer Instructions.
Information Encoding. Programming languages evolution. Assembler programming language.
(3 hours)

5. Algorithms.
Introduction to algorithms. Flowchart symbols. Representing algorithms through flowcharting and pseudcode. **(2 hours)**

6. Arithmetic, Relational, Boolean and Bitwise Operators and Expressions
Evaluating expressions. Relational operators. Assignment operators. Equality operators. Boolean data type. **(2 hours)**

7. Control Structures.
Sequential Structure. Selective Structure **if...then, if ... then... else, Case ....** Repetitive

Illustrating nested structures through practical examples.

9. Flowcharting Using Series of numbers. (1 hour).
Calculating the Sum/Product of a List of Numbers.

10. Functions: Flowchart/Pseudocode. (2 hour).
Why do we write functions. Function parameters. Steps to Writing a Function.

11. One dimensional arrays Flowchart/Pseudocode.
Declare, Initialize and Access Elements of one dimensional array. Traversing an array. Seaching an element. **(2 hours)**

12. Main operations using one-dimensional arrays.
Inserting an element. Deleting an element. Linear and Binary sorting algorithm. Parallel arrays. **(2 hour).**

13. Two dimensional arrays Flowchart/Pseudocode. (4 hours).
Declare, Initialize and Access Elements of two dimensional array.
Traversing an array. Seaching an element. Main operations on two dimensional arrays.

Laboratory Topics

1. Introducing Raptor.
Raptor is a simple-to-use problem solving tool that enables the user to generate executable flowcharts
2. Using Raptors symbols to generate flowcharts.
3. Using Raptors symbols to generate flowcharts of sequential structures. Assignment, Input, Output.
4. Using Raptors symbols to generate flowcharts of selective structures. Selection.
5. Using Raptors symbols to generate flowcharts of loop structures. Loop Control.
6. Building and calling functions inside Raptor flowchart.
7. Using Raptors symbols to generate flowcharts of nested iterative structures.
8. Using Raptors symbols to generate flowcharts of practical exercises with one dimensional arrays.
9. Using Raptors symbols to generate flowcharts of practical exercises with two dimensional arrays.
10. Laboratory work

Laboratory work grade: The Laboratory programs portion of your grade will be based on a laboratory work which will be assigned and submitted during the laboratory hour.
Attendance: All students must attend the labs 80% of lab hours.

Developed Skills

At the end of this course students will have good knowledge on solving computational problems through algorithms representing them in pseudocode or flowchart.

Literature

Textbook:
Arkaxhiu, Eqerem. Hyrje në Informatikë, SHBLU 2007. Written Lectures:
MUNGULI Brisilda. Website: http://informatika.fshn.edu.al/e-learning

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Physical Education

Subject activity	Introducing the Program	Games	Athletics	Gymnastics	Total
Student duties	1%	80%	10%	9%	100%
Class hours	0.3 hours	24 hours	3 hours	2.7hours	30hours
Learning Development:	First year; Semester I + II; 15 weeks; 1 Hours per Week				
Student Obligations:	Attendance 75%				
Valuation Mode: Wins Continuous Rating in%:	Wins Continuous Rating in%:				
Credits	1				
Exam form	Participation 10% Activation 60% Intermediate checks 20% Participation in sports activities 10% Maximum rating in%: Wins 85% -100% Minimum rating in%: Wins 40% -60%				

Subject description	To create a culture and model for an active life in students. -Educating and realizing human values. -Exacting with the feeling of collectivity, help and duty, facing challenges and competition. -Educating individual and group responses. -Argentation of students. - Provision of mental and physical health. - Improving the student's quality of li
Objectives	- Continuous improvement of functional physical quality (speed, strength, resistance, coordination ability and psycho-physical recovery capability). - Improvement of individual technique. - Improvement of tactical and theoretical elements in programmed sports.

Topics of the lectures

1. Knowledge of "Recognition of the program and anthropometric measurements".
2. "Free outdoor jogging" athletics.
3. Basketball "General knowledge of the basketball game. Regulation. Learning Game ".
4. Volleyball "Getting to know the main rules of the volleyball game".
5. Aerobic gymnastics "Complex of Exercised Exercises".
6. Aerobic gymnastics "Complex of exercises accompanied by music".
7. Football "Theoretical knowledge of the football game. Basic Rules ".

8. Handball "General knowledge of the game. Knowledge and learning of technical elements; Passing, waiting for the ball to go in and move. "
9. Handball "Variations of shooting from home and on the move with and without goalkeeper".
10. Basketball "Knowing the technical elements of the game. Teaching Game. Group Level Evidence ".
11. Basketball "Knowing and learning technical elements A- Catching the ball B-Possessing the ball from the table C- Rapture of the Ball. Teaching Game ".
12. Volleyball "Placement of the group on initial positions and positions on the field. (High, Medium, Low position) ".
13. Volleyball "Moving on the field (entering the free player on the second line in the game").
14. Athletics "Speeding (30-40-50m)
15. Athletics "Stability Running (6-8-10 min)".
6. Football "Ball Leadership and Binding With Legs".
17. Volleyball "Variations of Exercises and Ballroom Exercises Variations (Teaching Game)".
18. Volleyball "Variation of exercises for passing the ball at home and on the move".
19. Football "Taking and kicking the ball with different leg parts".
20. Football "Ball kick in the country and move in different distance".
21. Handball "Organizing defense, training in the area".
22. Basketball "Passing the ball. Types of passage. Use them in the game. Teaching Game ".
23. Basketball "Knowing and learning about field moves A- With ball B- Without ball (Learning Game)".
24. Volleyball "Variety of Individual Blocks and Groups (Teaching Game)".
25. Volleyball "Variation of Attack Training Exercises (4-3-2) Learning Game".
26. Basketball "Moving on the Field with Ball A-Driblimi, Types of Dribbling".
27. Basketball "Ball Shootout. Knowing and learning the shooting technique from: A-Place B-In-motion (Lesson game) ".
28. Football "Game on diminished territory (2: 2; 3: 3 etc.)".
29. Football "Blow in the door with variants. Defense game against attack.

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

PHYSICS

Subject activity	Lectures	Exercises	Laboratories	Practice	Total
Student duties	Not compulsory	75%	100%	100%	0
Class hours	60	60	0	0	120
Individual studies	130				
Teaching language	Albanian				
Subject typology / Type of the subject/ Subject code	A / Compulsory/				
Ethical code	Referred to Ethical code of UT, approved by Decision No. 12, date 18.04.2011				
Evaluation method (ESE/TSE)	ESE (EXAM)				
Credits	**10**				
Lesson form	Year I, Semester I+II, 30 weeks: 2 hours of lectures / week, 2 hour of exercises / week.				
Exam form	Written and oral. The written exam is eliminatory. 40-45 points grade five and with every ten points, the grade increases by one grade.				
End Semester Evaluation (ESE)	Presence and active participation	5 %			
	Midterm control	10 %			
	Final exam	85 %			
	Total	**100%**			
Professor	**Prof. As. Dr. Polikron Dhoqina & Dr. Rudina Osmanaj** **Prof. Asoc. Lindita Hamolli & Dr. Erjon Spahiu**				

Basic concepts	It is the basic discipline of training Informatics students. The first part of this course introduces the student to the phenomena and basic laws of classical physics, especially mechanics, waves, thermodynamics and electromagnetism. The second part of this course introduces the student to the phenomena and basic laws of modern physics. In particular, the basics of Quantum Mechanics and its applications to atoms, conductors, semiconductors, semiconductor and devices with semiconductors. This part of the course not only contributes to the overall formation of computer science students, but also helps them to acquire new technologies, in particular the next generation of quantum computers. All chapters of the course are accompanied by relevant exercises, where the student applies in different situations the theoretical knowledge gained in lectures.
Objectives	The course aims at recognizing and using the basic concepts and laws of General Classical Physics and Modern General Physics. It also aims to know the laws of some specific phenomena related to computer technology.
Foreknowledge	Basic knowledge of high school in physics and in mathematics.

Topics of the Lectures

Part I, Classical Physics

1.Kinematics and Newton's laws. (4 hours)
Introduction. Speed and acceleration. Straight line movement. Movement in the plan. Movement of shells. Newton's laws. Mass and weight. Applications of Newton's laws. Page 1 - 21

2. Laws of conservation (3 hours)
Work performed by a force. Power. Potential energy. Kinetic energy. Energy conservation. Center of mass.

Movement of the center of mass. Shocks. Page 30 - 48

3. Rotational movement (3 hours)

Rotational motion. Angular velocity. Acceleration and centripetal force. Orbital motion. Gravity. Moment of inertia and moment of force. Kinetic energy of rotation. Moment of impulse. Page 57 - 73

4. Kinetic theory of gases (4 hours)

Introduction. Ideal gas laws and absolute temperature. Kinetic theory of pressure and temperature. Specific heat. The first principle of thermodynamics. Statistical distribution of Maksuell-Boltzman. The second principle of thermodynamics. Karno Cycle. Entropy and its statistical meaning. Page 82 - 103

5. Vibrations and waves (4 hours)

Introduction. The oscillations of a spring. Vibration energy. Wandering waves on a string. Wave energy transport. The principle of superposition and interference. Light interference. Diffraction. Stable waves. Page 108 - 136

6. Electrostatics (4 hours)

Introduction. Coulomb's law. The principle of superposition and electric field. Potential energy and electric potential. Electromotive force and electric capacity. Page 148 - 162

7. Electricity (4 hours)

Introduction. Electricity. Omit's law. Resistance. Kirkoff rules. Power consumed in resistance. Capacitor charging. Page 171 - 186

8. Magnetic field and electromagnetic waves (4 hours)

Introduction. Force on current conductors. The force on a moving load and the Hall effect. Electromagnetic waves. The nature of light. Page 191 - 203

Part II, Modern Physics

9. The Beginning of the Quantum Physics (4 hours)

Introduction. Blackbody Radiation. The Photoelectric Effect. Further Evidence for the Photon Theory. X- Rays. Compton effect. Page 209 - 226

10. Atom Models 4 hours)

Introduction. The Rutherford Model. The Spectrum of Hydrogen. The Bohr Atom. The Franck-Hertz Experiment. Page 230 - 242

11. Fundamental Principles of Quantum Mechanics (4 hours)

Introduction. De Broglie's Hypothesis and Its Experimental Verification. Nature of wave. The uncertainty principle. Physical Origin of the Uncertainty Principle. Matter Waves and the Uncertainty Principle. Velocity of the Wave Packet: Group Velocity. The Principle of Complementarity. Page 246 - 261

12. An Introduction to the Methods of Quantum Mechanics (4 hours)

Introduction. The Schrodinger Theory of Quantum Mechanics, Application of the Schrodinger Theory. Outline of the Solution of the Schrodinger Equation for the H Atom. Physical Significance of the Results. Space Quantization: The Experiments. The Spin. Some Features of the Atomic Wavefunctions. The Periodic Table, Page 266 - 314

13. Crystal Structures and Bonding in Solids (5 hours)

Introduction. Crystal Structures. Crystal Bonding. Classical Free Electron Model. Quantum-Mechanical Free Electron Model.. Fermi-Dirac Statistics. Page 318 - 354

14. Band Theory of Solids. (5 hours)

Introduction. Bloch's Theorem. The Kronig-Penney Model. Tight-Binding Approximation. Conductors, Insulators, and Semiconductors. Effective Mass. Holes. Intrinsic Semiconductors. Extrinsic or Impurity Semiconductors. Electrical Conductivity of Semiconductors. Photoconductivity. Page 359 - 417

15. Semiconductor Devices (4 hours)

Introduction. Metal-Metal Junction: The Contact Potential. The Semiconductor Diode. The-Bipolar Junction Transistor (BJT). Field-Effect Transistors (FET). Page 423 - 456

Exercise topics

1. Kinematics and Newton's laws	(6 hours)
2. Laws of conservation	(6 hours)
3. Rotational movement	(4 hours)
4. Kinetic theory of gases	(4 hours)
5. Vibrations and waves	(4 hours)
6. Electrostatics	(4 hours)
7. Electricity	(4 hours)
8. Magnetic field and electromagnetic waves	(4 hours)
9. Beginnings of quantum mechanics	(4 hours)
10. Atom models	(4 hours)
11. Basic principles of quantum mechanics	(4 hours)
12. Introduction to methods of quantum mechanics	(4 hours)
13. Crystalline structure and theories of free electrons in solids	(4 hours)
14. Zonal theory of rigid bodies. Semiconductors	(2 hours)
15. Semiconductor devices	(2 hours)

Skills given (Student learning outcomes)

The student receives a general training in the disciplines of Classical Physics and Modern Physics. He feels proficient in using the concepts and laws of classical mechanics, thermodynamics, optics, electromagnetism, quantum mechanics, the physics of atoms, crystals, semiconductors, and semiconductor devices.

Literatura

Required Literature:

B. Duka, P. Dhoqina, "Fizika për studentët e informatikës" Tiranë, text adaptation: "Physics for Computer Science students", 1997, *authors:* Narciso Garcia, Arthur Damask, Steven Schwarz.

Auxiliary literature:

F. Sinoimeri, Z. Mulaj, B. Duka, Fizika 1 , botim i SHBLU, Tiranë 2004
P. Dhoqin: Elektriciteti dhe Magnetizmi , Tiranë, 2008
P. Dhoqina: Problema të Fizikës, Tiranë, 2010

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Algebra

Teaching Activities	Lectures	Exercises	Lab. Work	Total
Frequentation Requirement	Non mandatory	100%	0	0
Teaching Hours	30	30	0	60
Individual Study Hours	60			
Language of Instruction	Albanian			
ECTS Credits	5			
Teaching Methods	First Year, First Semester, 15 weeks: 2 lecture hours, 2 exercise hours,			
Assessment	Participation and activation	20%		
	Laboratory	0%		
	Assignments/Project	10%		
	Final exam	70%		
	Total	**100%**		
Main Instructor				

Course Description	The place that holds the subject in holistic formation: The discipline of the overall program formation Prerequisites for student acquisition of subject: Mathematical analysis
Objectives	The objective of this course is to equip students with basic algebraic culture and at the same time prepare them with the necessary algebraic knowledge for traditional applications of algebra in computing. Understanding the students with the basic concepts and techniques of algebra, with three main purposes: first, to familiarize themselves with the algebraic capacities in formalizing and solving some of its classical problems; Secondly, to have the equipment needed for its use in non-algebraic maths and maths, and finally to create the necessary basis for further studies in other vocational training subjects such as cryptography coding theory.
Preliminary Requirements	Topics to be covered in Theoretical and Practical Formation

Lecture Topics

Induction
Their Equivalence.
Factoring. Euclid's Algorithm, Partition Theorem,
The Biggest Common Allocation. Single factorization.
Simple Numbers, Congregations, Congregation Classes.
Rings and fields. Definitions, Ring rings Zm.
The square matrix ring of order n.
Complex numbers.
The trigonometric shape of the complex number.
The root of complex numbers.
Elements that have inverse.
The order of the elements of a ring.
Homomorphisms of the rings
Farm Theorem and Euler Theorem.
Element Order, Farm Theorem, Theorem's Theorem,

Groups. Definition and First Attributes, Homomorphisms of Groups, Finite Groups, Subgroups, Cyclic Groups, Neighboring Classes. Lagrange theorem. Exponent of the abelian group.
Subunits and subfunctions.
The elements of a unitary ring that have inverse.
Polynomials. Construction of polynomial ring, Polynomial roots, Partition theorem.
The largest common participant and the Bezu draw for polynomials.
The polynomial breakthrough.
Incongruous polynomials and canonical decomposition of a polynomial.
Polynomials with complex coefficients, Polynomials with real coefficients
Rings and polynomials mod f (x).
Polynomials on finite fields.
Chinese Waste Theorem.
The production of rings and the φ-function of the Eiler. Chinese Residue Theorem for Polynomials.
Primitive roots and cyclic groups.
Primitive roots mod n.
Primitive polynomials. Cyclical groups subgroups.
Determinants. Defining the Definitor.
Specifier properties.
Reverse matrix.
Kramer formulas.
Cryptography,
Hill figures,
RSA figures.
Linear spaces.
Definition of linear spaces, linear subsurface, linear dependence, dimensions and basis of space. Cartesian production space.
Amount and line spacing.
Base Matrix Matrix.
Basic properties, examples.
Linear dependence, bases, dimensions, coordinates.
The amount of subsurface.
Matrix Linear Equations Linear Matrix Systems.
Systems of linear equations
Homomorphisms of linear spaces, basal meanings.
Homomorphism matrices, Isomorphisms, case and dimensional spaces.
Homogeneous systems.

Developed Skills

Upon successful completion of this course, students will have an understanding of the basic areas of algbra including knowledge discovery, decision making, and learning – as well as their applications.

Literature

Auxiliary Literature:

1. K.Filipi- "Algebra and Geometry", ShBLU, 1988
2. K.Filipi, A.Baxhaku- "Exercises and Problems of Algebra and Geometry", SHBLU, 1997
3. L.Childs- "A Concrete Introduction to Higher Algebra", Springer-Verlag New York Inc. 1983

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

MATHEMATICS

ACTIVITY	Lecture	Seminar	Laborator	Practice	Total
Attendance	Not compulsory	75%	-	-	-
Class hours	56	56		-	112
Individual studies	112.5				
Language	albanian				
Evaluation	exam				
ETCS	12				
Program schedule	First year, semester I+II, 14 weeks: 2 lecture hours, 2 seminar hours per week				
Exam	Written form				
Evaluation, grades, exam	Class activisation	10%			
	Midterm				
	Homework				
	Laborator				
	Practice				
	Final exam	90%			
	total	**100%**			
Lecturer	**Dr. Besiana Çobani** *tel:, e-mail: besiana.hamzallari@fshn.edu.al*				

Overview	The course is focused in topics on calculus.The basic mathemamatical concepts are studied such as the limit of a function, its continuity and differentiability. The multivariable calculus is developed in the second semester.
Objectives	The main goal of the course is to provide the essential knowledge of the calculus to the student. Since the student of informathics is more interested in the applications of the mathematical concepts, a reasonable part of the class hours is left to the examples that represents the connection between theory and application.
Prerequisites	Good knowledge of the high school mathematical curiculla

Topics

1. Sets. Operation with sets. Infinimium and supremium. 25-51 [1]
2. Functions. Limit and continuity. 72-116 [1]
3. Infinitesimals 126-132 [1]
4. Important theorems on a continuous function in a closed interval 137-155[1]
5. Differentation. 185-225 [1]
6. Ferma theorem, Role theorem, Lagrange theorem 227-240 [1]
7. L'Hapital Rule. Taylor formula 243-260 [1]
8. Antiderivative. 284-300 [1]
9. Integral by parts, by substitution. [1]
10. Integration of Trigonometric functions[1]
11. Limit of a sequence.
12. Numerical series. Test of comparison, test of limit and integral test. 1-15 [2]
13. Ratio test and root test. Absolute convergence 17-30 [2]
14. Power series 51-66 [2]
15. Taylor expansion of a function 68 -90 [2]
16. Multivariable function. Limit and continuity 123-149 [2]
17. Partial derivative. Minima and maxima 160 -201[2]

18. Double integral in rectangular domain[2]
19. Integration by substitution. Polar coordinates 301 -331 [2]

Learning outcomes

This course enables the student to deal and solve basic calculus exercises and problems. The material helps the student to use mathematical tools in their own proffesional problem. Also, the logic deduction, which is in the base of all the course, motivates the student to cope with problems which arise in his everyday life.

Literature

Compulsory literature

1. **AnalizaMatematike I, II.** (*Teori*). Kristaq Gjino Per studentet e deges Matematike, Fizike, Informatike.
2. **AnalizaMatematike I, II.**(*Ushtrime*) Kristaq Gjino Per studentet e deges Matematike, Fizike, Informatike **)**
3.

Auxiliary literature

1. J. Stewart Calculus Early Transcedentals Sixth Edition Thomson Brooks, 2008
2. **Analizë Matematike 1 Ushtime e probleme të zgjidhura. Bashkim Gazidede**
3. **Analizë Matematike 1 Ushtime e probleme të zgjidhura. Bashkim Gazidede**

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Data structure in C

Teaching Activities	Lectures	Exercises	Lab. Work	Practice	Total
Frequentation Requirement	80%	75%	100%		0
Teaching Hours	60	45	30	0	135
Individual Study Hours	165				
Language of Instruction	Albanian				
Evaluation method	Exam				
ECTS Credits	11				
Teaching Methods	Year I, Semester I, 15 weeks: 2 hours lectures, 1 hours seminars, 1 hour laboratory/week Semester I, 15 weeks: 2 hours lectures, 2 hours seminars, 1 hour laboratory/week				
Scoring Details	Only writing, 40% of points obligatory, 45-50 points mark five, every 10 points evaluation increases one mark.				
	Midterm	20%			
	Laboratory	10%			
	Final exam	70%			
	Total	100%			
Main Instructor	Dr.Denis SAATÇIU, *tel:++355672197834, e-mail:denis.saatciu@fshn.edu.al*				

Course Description	The course teaches teoritical and practical concepts of a programming language, making possible writing structured programs. The subject teaches C programming too young developers. Programming concepts on variables, loops, data structures, etc. are learned. The subject is focused on building a correct C program through learning structured programming techniques.
Objectives	Understand the process of automatisation. Learn the enviroinment of a programming language. Learn how to analise a simple problem. Learn how to use the different types of data structure Learn how to choose the correct data structure
Preliminary Requirements	None
Lecture Topics	
Semester I Introduction on structured programming. Basics of C environment. History of programming techniques. Strructured programming. Introduction to C language. Steps followed from writng the code to its execution. General review of the structure of a program written in C (2 hours) Introduction to programming. Database types. Input Output instruction. Datatypes in C. Variables, constants. Instructions printf, scanf to communicate with the console (2hours). Decision inctructions if/else and operators. Instructions If-Else. Instruction Switch. (2hours). While loop structure. Shorted operators. Increase and decrease operators. Operators, shorted operators, priority table of operators. Loop concept. While loop. (2 hours).	

For loop. Switch-multiple choice structure. For loop. Nested For loop (2 hours).
Do While Loop. Instructions break and continue. Logic operators. Do-While loops. Instructions Break, Continue, Exit. (2 hours).
Functions. Definitions, prototypes. Call by value. Declarations, function prototype, usage in the main function. (2 hours).
Called by address functions, procedures, functions wihtout parameters. Default arguments. Specific function types. (2 hours).
One dimensional arrays. Declaration and usage. Declaration and storage. Reading and writing methods. (2 hours).
Algorithms with onedimensional arrays, sorting, searching. Algorithms with onedimensional arrays such as sorting(optimizing instruction number), searching an element in the array, adding/removing an element in the array, adding/removing an element in the k-th position in the array, etc. (2 hours).
Multidimensional arrays. Declaration and usage. Declaration and storage. Reading and writing methods. (2 hours).
Algorithms with multidimensional arrays. Algorithms with multidimensional arrays combined with onedimensional arrays. (2 hours).
Strings. String manipulation functions. Examples. Declaration and storage of a string. String manipulation. (2 hours).
Structures. Definition. Declaration of structures in C. Accessing structure elements. (2 hours)
Arrays with structures. Applications of structure data types on problems solvable with arrays. (2 hours)

Semester II

Introduction to data structures. Data structure role. Variables. Arrays. Conditions. (2 hours)
Recursivity. Recursivity concept. Recursivity functions. (2 hours)
Recursivity, using arrays. Recursivity applications, aspects, recursivity exercises with arrays (2 hours)
Files characteristics, text files. Files. Reserving. Opening/Closing the file. Usage of text files. (2 hours)
Usage of text files. Binary files. Differences between text and binary files. Text files with records. Copying text files. Sorted files. Merging teo sorted files. Direct access files. (2 hours)
Linked list, characteristics. Linked list. Main algorithms on linked lists. (2 hours)
Sorted linked list, creation and usage. Specific typed of linked lists. Usage. (2 hours)
Stacks and Queues. Characteristic and construction. Usage of linked list. Stacks, queues. Features and structure. (2 hours)
Trees. Construction and characteristic. Trees. Structure of tree elements. Tree creation. Accessing elements on sorted trees. Accessing algorithms. (2 hours)
Binary trees. Binary search trees. Heaps. Binary trees. Binary tree structure. Accessing algorithms. Searching in binary trees. Heaps. (2 hours)
B trees, M trees. Arithmetic expressions. Expression evaluation. (2 hours)
Hash methods. Data updating. Hash method. Hash functions. Collisions. (2 hours)
Sorting. Array sorting. Insertion sort, bubble, shell sort etc. (2 hours)
Sorting (continue). Sorting using trees. Quick sort. (2 hours)
Review. (2 hours)

Seminar Topics

Semester I

Introduction on structured programming. Basics of C environment. Exercises: History of programming techniques. Strructured programming. Introduction to C language. Steps followed from writng the code to its execution. General review of the structure of a program written in C (2 hours)
Introduction to programming. Database types. Input Output instruction. Exercises: Datatypes in C. Variables, constants. Instructions printf, scanf to communicate with the console (2hours).
Decision inctructions if/else and operators. Exercises: Instructions If-Else. Instruction Switch. (2hours).
While loop structure. Shorted operators. Increase and decrease operators. Exercises: Operators, shorted

operators, priority table of operators. Loop concept. While loop. (2 hours).
For loop. Switch-multiple choice structure. Exercises: For loop. Nested For loop (2 hours).
Do While Loop. Instructions break and continue. Logic operators. Exercises: Do-While loops. Instructions Break, Continue, Exit. (2 hours).
Functions. Definitions, prototypes. Call by value. Exercises: Declarations, function prototype, usage in the main function. (2 hours).
Called by address functions, procedures, functions wihtout parameters. Exercises: Default arguments. Specific function types. (2 hours).
One dimensional arrays. Declaration and usage. Exercises: Declaration and storage. Reading and writing methods. (2 hours).
Algorithms with onedimensional arrays, sorting, searching. Exercises: Algorithms with onedimensional arrays such as sorting(optimizing instruction number), searching an element in the array, adding/removing an element in the array, adding/removing an element in the k^{th} position in the array, etc. (2 hours).
Multidimensional arrays. Declaration and usage. Exercises: Declaration and storage. Reading and writing methods. (2 hours).
Algorithms with multidimensional arrays. Exercises: Algorithms with multidimensional arrays combined with onedimensional arrays. (2 hours).
Strings. String manipulation functions. Examples. Exercises: Declaration and storage of a string. String manipulation. (2 hours).
Structures. Definition. Exercises: Declaration of structures in C. Accessing structure elements. (2 hours)
Arrays with structures. Exercises: Applications of structure data types on problems solvable with arrays. (2 hours)

Semester II

Introduction to data structures. Exercises: Data structure role. Variables. Arrays. Conditions. (2 hours)
Recursivity. Exercises: Recursivity concept. Recursivity functions. (2 hours)
Recursivity, using arrays. Exercises: Recursivity applications, aspects, recursivity exercises with arrays (2 hours)
Files characteristics, text files. Exercises: Files. Reserving. Opening/Closing the file. Usage of text files. (2 hours)
Usage of text files. Binary files. Differences between text and binary files. Exercises: Text files with records. Copying text files. Sorted files. Merging two sorted files. Direct access files. (2 hours)
Linked list, characteristics. Exercises: Linked list. Main algorithms on linked lists. (2 hours)
Sorted linked list, creation and usage. Exercises: Specific typed of linked lists. Usage. (2 hours)
Stacks and Queues. Characteristic and construction. Exercises: Usage of linked list. Stacks, queues. Features and structure. (2 hours)
Trees. Construction and characteristic. Exercises: Trees. Structure of tree elements. Tree creation. Accessing elements on sorted trees. Accessing algorithms. (2 hours)
Binary trees. Binary search trees. Heaps. Exercises: Binary trees. Binary tree structure. Accessing algorithms. Searching in binary trees. Heaps. (2 hours)
B trees, M trees. Arithmetic expressions. Exercises: Expression evaluation. (2 hours)
Hash methods. Exercises: Data updating. Hash method. Hash functions. Collisions. (2 hours)
Sorting. Exercises: Array sorting. Insertion sort, bubble, shell sort etc. (2 hours)
Sorting (continue). Exercises: Sorting using trees. Quick sort. (2 hours)
Review. (2 hours)

Laboratory Topics

Semester I

Introduction on structured programming. Basics of C environment. Exercises: History of programming techniques. Strructured programming. Introduction to C language. Steps followed from writng the code to its execution. General review of the structure of a program written in C (2 hours)
Introduction to programming. Database types. Input Output instruction. Exercises: Datatypes in C. Variables, constants. Instructions printf, scanf to communicate with the console (2hours).

Decision inctructions if/else and operators. Exercises: Instructions If-Else. Instruction Switch. (2hours).
While loop structure. Shorted operators. Increase and decrease operators. Exercises: Operators, shorted operators, priority table of operators. Loop concept. While loop. (2 hours).
For loop. Switch-multiple choice structure. Exercises: For loop. Nested For loop (2 hours).
Do While Loop. Instructions break and continue. Logic operators. Exercises: Do-While loops. Instructions Break, Continue, Exit. (2 hours).
Functions. Definitions, prototypes. Call by value. Exercises: Declarations, function prototype, usage in the main function. (2 hours).
Called by address functions, procedures, functions wihtout parameters. Exercises: Default arguments. Specific function types. (2 hours).
One dimensional arrays. Declaration and usage. Exercises: Declaration and storage. Reading and writing methods. (2 hours).
Algorithms with onedimensional arrays, sorting, searching. Exercises: Algorithms with onedimensional arrays such as sorting(optimizing instruction number), searching an element in the array, adding/removing an element in the array, adding/removing an element in the k-th position in the array, etc. (2 hours).
Multidimensional arrays. Declaration and usage. Exercises: Declaration and storage. Reading and writing methods. (2 hours).
Algorithms with multidimensional arrays. Exercises: Algorithms with multidimensional arrays combined with onedimensional arrays. (2 hours).
Strings. String manipulation functions. Examples. Exercises: Declaration and storage of a string. String manipulation. (2 hours).
Structures. Definition. Exercises: Declaration of structures in C. Accessing structure elements. (2 hours)
Arrays with structures. Exercises: Applications of structure data types on problems solvable with arrays. (2 hours)

Semester II

Introduction to data structures. Exercises: Data structure role. Variables. Arrays. Conditions. (2 hours)
Recursivity. Exercises: Recursivity concept. Recursivity functions. (2 hours)
Recursivity, using arrays. Exercises: Recursivity applications, aspects, recursivity exercises with arrays (2 hours)
Files characteristics, text files. Exercises: Files. Reserving. Opening/Closing the file. Usage of text files. (2 hours)
Usage of text files. Binary files. Differences between text and binary files. Exercises: Text files with records. Copying text files. Sorted files. Merging teo sorted files. Direct access files. (2 hours)
Linked list, characteristics. Exercises: Linked list. Main algorithms on linked lists. (2 hours)
Sorted linked list, creation and usage. Exercises: Specific typed of linked lists. Usage. (2 hours)
Stacks and Queues. Characteristic and construction. Exercises: Usage of linked list. Stacks, queues. Features and structure. (2 hours)
Trees. Construction and characteristic. Exercises: Trees. Structure of tree elements. Tree creation. Accessing elements on sorted trees. Accessing algorithms. (2 hours)
Binary trees. Binary search trees. Heaps. Exercises: Binary trees. Binary tree structure. Accessing algorithms. Searching in binary trees. Heaps. (2 hours)
B trees, M trees. Arithmetic expressions. Exercises: Expression evaluation. (2 hours)
Hash methods. Exercises: Data updating. Hash method. Hash functions. Collisions. (2 hours)
Sorting. Exercises: Array sorting. Insertion sort, bubble, shell sort etc. (2 hours)
Sorting (continue). Exercises: Sorting using trees. Quick sort. (2 hours)
Review. (2 hours)

Developed Skills

In the end of the course, the student will have knowledge on:

Basic concepts of programming, such as variables, loops, functions etc.
Data structures, such as arrays one and multi dimensionals, strings, structures, files, linked lists, queues, stacks, heaps, trees.
Writing structured programs through effective algorithms.

Literature
Writing lectures:
NINKA, Ilia. Gjuha C & Programim i Strukturar.
NINKA, Ilia. Struktura të dhënash nën C.

Recommended leterature:
1. http://www.cs.sunysb.edu/~skiena/214/lectures/
2. Michael T. Goodrich Roberto Tamassia David M. Mount
"Data Structures and Algorithms in C++ "

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

English 1

Teaching Activity	Lectures	Exercises	Laboratories	Practical	Total
Forcing a student	0%	80%	0%	0%	0
Classes		60	0	0	60
Individual Study	90				
Language learning development	English				
Method of settlement	Assessment continued				
Credits	6				
Development of Learning	Year I, Semester I, 15 weeks: 2 hour seminar / week Year I, Semester II, 15 weeks: 2 hour seminar / week				
Exam Development					
Assessment, Grading VORDRUCK	Activation	5%			
	Assignments	10%			
	Control 1	15%			
	Control 2	70%			
	Total	**100%**			
Responsible Course	**Ass Arjan SHUMELI arjan.shumeli@fshn.edu.al**				

Course Description	Compulsory subject Discipline general training
Objectives	To update the knowledge gained by students in high school and present them with relevant terminology in the field of ICT through practicing interpersonal skills, as well as strengthening and promoting sustainable structural knowledge, the lexical grammar students.
Prior knowledge	According CEFL B2 (Common European Framework of Languages)

The theme of the seminars

1. **Computers.** *Vocabulary:* Uses of computers. Studying IT. *Reading:* Computers in everyday life. *Grammar:* Present Simple. Adverbs of frequency. *Speaking:* Meeting new people. *Writing:* Personal use of computers. **(4 or h).**
2. **Parts of computer.** *Vocabulary:* Peripherals. Types of computer. The system unit. *Reading:* A robot that helps you. *Grammar:* Comparatives. Superlatives. *Speaking:* Buying and Selling a computer. *Writing:* A for and Against an essay. **(4 or h).**
3. **Computer Devices.** *Vocabulary:* Input and output devices. Storage devices. *Reading:* Printers, Our friends? *Grammar:* Relative clauses. For + gerund. *Speaking:* Describing an object. *Writing:* describing a gadget. **(4 or h).**
4. **Technical Support.** *Vocabulary:* Computer Problems. *Reading:* Rules for a good Customer Service. *Grammar:* Questions. *Speaking:* Customer Complaints. *Writing:* A Letter of Complaint. **(4 or h).**
5. **The Internet of.** *Vocabulary:* The alphabet. Domain names. Email addresses. *Reading:* Work Abroad experience. *Grammar:* Present Simple and Present Continuous. *Speaking:* Setting up a business. *Writing:* An informal email. **(4 or h).**
6. **The Internet OF.** *Vocabulary:* netiquette. *Reading:* Forum rules. *Grammar:* Simple Past. *Speaking:* A personal story. *Writing:* A blog entry. **(4 or h).**
7. **Entertainment.** *Vocabulary:* Social Networks. Types of videogames. *Reading:* The Other Side of Social Networks. *Grammar:* Future Simple. Be going to. *Speaking:* Presentation of a videogame. *Writing:* A videogame review. **(4 or h).**

8. **Dangers.** *Vocabulary:* Computer Crimes and Dangers. *Reading:* The Top Five Computer Crimes. *Grammar:* Should / shouldn't. Must / Have to. *Speaking:* A bad experience. *Writing:* Asking for / GIVING ADVICE. **(4 or h).**
9. **Office Software.** *Vocabulary:* Word processors. Spreadsheets. *Reading:* Commands in Microsoft Word. *Grammar:* imperatives. Let's and please. *Speaking:* Introducing date. *Writing:* Giving instructions. **(4 or h).**
10. **Computer Types (4 or h).**
11. **Programming Languages. (Or 4 h).**
12. **The Keyboard Part 1 (or 4 h).**
13. **The keyboard Part 2 (or 4 h).**
14. **IT Careers (4 or h).**
15. **People in IT (or 4 h).**

Skills which equip student

This course teaches students how to use the English language can be oriented in computing and information technology.

Literature

Recommended Literature:

1. Bel N. Pascual Lence, Verónica Real Morte. Basic English for Information Technology,2012.

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Second Year Courses

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

English 2

Teaching Activity	Lectures	Exercises	Laboratories	Practical	Total
Forcing a student	0%	80%	0%	0%	0
Classes		60	0	0	60
Individual Study	90				
Language learning development	English				
Method of settlement	Assessment continued				
Credits	**6**				
Development of Learning	Year II Semester, 15 weeks: 2 hour seminar / week Year II Semester II, 15 weeks: 2 hour seminar / week				
Exam Development					
Assessment, Grading VORDRUCK	Activation	5%			
	Assignments	10%			
	Control1	15%			
	Control 2	70%			
	Total	100%			
Responsible Course	Ass Arjan SHUMELI arjan.shumeli@fshn.edu.al				

Objectives	To update the knowledge gained by students in high school and present them with relevant terminology in the field of Information Technology through practicing interpersonal skills, as well as strengthening and promoting sustainable structural knowledge, the lexical grammar students.
Prior knowledge	According CEFL B2 (Common European Framework of Languages) Knowledge gained in the course of the first year

The theme of the seminars

1. Computer Users. Language Work: Revision: Past Simple and Present Perfect. Speaking: Exchanging information. Writing: Writing a brief description. (2 or h).
2. Computer Architecture. Language Work: Describing how an item functions. Prepositions of place. Speaking: Exchanging Technical Information. Writing: Sequencing instructions. (2 h).
3. Computer Applications. Language Work: Present liabilities. Speaking: Describing a process. Writing: Describing a process. (2 h).
4. Peripherals. Language Work: Revision: Comparison and contrast. Writing: Comparing and contrasting Describing function. (2 h).
5. Interview: Former Student. Language Work: Revision: Past Simple Word Study questions: an up girl - up verbs. Writing: Describing function. (2 h).
6. Operating Systems. Language Work: - ing or noun form and After prepositions. Speaking: Matching text and diagram predictions. (2 h).
7. Graphical User Interface. Language Work: , Perm. Reading: Reading Diagrams. Speaking: Providing explanations. (2 h).
8. Applications Programs. Language Work: Instructions and complex instructions. Reading: Note-Taking. Speaking: Exchanging information. Writing: Making Recommendations. (2 h).
9. Multimedia. Language Work: - ing clauses: cause and effect. Reading: locating information in the diagram and text. Speaking: Providing

explanations. Writing: Describing a process. (2 h).

10. Interview: Computing Support. Language Work: if-sentence Types 1 and 2 Word Study: noun + noun compounds. Speaking: Giving instructions. Reading: Matching diagram and ELITE output. (4 h).

11. Networks. Language Work: Relative clauses with a participle. Reading: Matching text and diagram. Speaking: Providing explanations. Writing: Describing Advantages and disadvantages. (2 h).

12. The Internet. Language Work: warnings. Writing: Writing a newsgroup contribution. Reading: Computer Mediated Communication. (2 h).

13. The World Wide Web. Language Work: Time clauses. Writing: Describing a process. Speaking: Information transfer. (Or 4 h).

14. Websites. Language Work: Giving Advice. Reading: Understanding the writer's Purpose. Speaking: Exchanging information. Writing: Evaluating. (2 h).

15. Encore: Webpage Creator. Language Work: Word Study: definitions and collocations. Speaking: Exchanging information. Writing: Advising. (2 h).

16. Communications Systems. Language Work: Predictions: certainty expressions. Reading: Reading for specific information. Speaking: Exchanging information to complete a diagram. Writing: Describing a system. (2 h).

17. Computing Support. Language Work: Diagnosing a fault and GIVING ADVICE. Speaking: Giving Advice on Technical problems. Writing: Reporting a problem. (2 h).

18. 1 Data Security Language Work: Cause and effect cause to / make; causative verbs; energy and - en verbs. Reading: Scanning. Speaking: Exchanging information. Writing: Explaining a computer crime. (4 h).

19. 2 Data Security Language Work: Cause and effect using Allow and Prevent links. Reading: Reading a table. Speaking: Exchanging explanations. Writing: Describing how a system Operates. (2 h).

20. Interview: The ex-hacker. Language Work: Phrasal verbs Word Study: Semantic Groups. Speaking: Role play. Writing: Writing a short news item. (2 or h).

21. Software engineering. Language Work: Revision: if X, Y will crack until, DO WHILE. Reading: Reading a table. Speaking: Exchanging information and options. Writing: Describing Advantages and disadvantages. (4 h).

22. People in Computing. Language Work: Requirements: need to, have to, must, eu + Essential / Critical. Reading: Reading and Note Taking. Speaking: Asking questions TARGETED. Writing: Writing a CV (2 h).

23. Recent Development in IT. Language Work: Ability: Can, Could, Able to swear. Reading: Reading and Note Taking. Speaking: Making a presentation. Writing: Writing a report. (2 h).

24. The future of IT. Language Work: Predictions: Future Perfect and It in subject position. Reading: Reading and Note Taking. Speaking: Persuading others to your point f view. Writing: Writing a summary. (4 h).

25. Interview: Electronic Publishing. Language Work: Emphasizing: cleft sentence Word Study: prefixes; - ise werbs. Reading: Reading and Note Taking. Speaking: Planning group presentation. Defending a Decision. (2 h).

Skills which equip student

This course prepares students to learn how to use the English language oriented computing and information technology.

Literature

Recommended Literature:

1. Eric Glendinning, John McEwan. Oxford English for Information Technology, Second Edition, 2002.

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

APPLIED MATHEMATICS

Subject activity	Lectures	Exercises	Laboratories	Practice	Total
Student duties	Not mandatory	75%	0	0	0
Class hours	60	60	0	0	120
Individual studies					
Language of instruction	Albanian				
Evaluation method	Written exam				
ECTS Credits	11				
Teaching methods	Second year, two semesters, 30 weeks: 2 lecture hours, 2 seminar hours				
Scoring details	Only written exam, 45% of points required, 45-50 points for grade 5, every 10 additional points grading is increased by one.				
Assessment, Note, Exam	Presence and active participation	10%			
	Midterm control	40%			
	Course work				
	Laboratories				
	Practice				
	Final control	50%			
	Total	**100%**			
Main instructor	**Eva Noka,** *tel:+355692082837, e-mail: eva.jani@fshn.edu.al*				

Subject description	The subject consists of 2 parts which are developed in parallel: Graph Theory and Networks, Probability and Statistics, disciplines pertaining to Applied Mathematics
Objectives	Provide basic knowledge on graphs and networks, their structure, properties, related problems and algorithms for solving them. Provide mathematical knowledge of probability concepts and basics of statistics, for a sound and application-oriented mathematical formation.
Preliminary requirements	Basic formation in Algebra, Mathematical Analysis and Data Structures

Topics of the lectures

Graphs, definitions and fundamental concepts. (3 hours). Variations of graphs. Vertex Degrees. Paths and Cycles. Different representations of graphs.
Trees, Arborescences and their characterizations(4 hours). Their main properties. Graph traversal: Breadth-First Search (BFS) and Depth-First Search (DFS). Minimum Spanning Tree. Kruskal's and Prim's Algorithms.
Vertex-connectivity and edge-connectivity (2 hours). Separating sets, cutvertices and bridges. The relationship between vertex-connectivity, edge-connectivity and minimum degree. Menger's Theorem.
Eulerian cycles and Hamiltonian cycles (2 hours). Cyclomatic and cocyclomatic numbers.
Planar graphs (1 hour). Understanding plane graphs and some of their properties. Topological characterization of planar graphs.
Basic knowledge on matching (5 hours). Bipartite graph matching and applications. Maximum matching algorithm in bipartite graphs. Minimum weighted bipartite matching and Hungarian algorithm.
Graph coloring and applications (1 hour). The chromatic number. A greedy algorithm for the graph coloring.
Shortest path problem from a source node (4 hours). The optimality conditions. Dijkstra's and Bellman-Ford algorithms. All-pairs shortest path problem. The optimality conditions. Floyd-Warshall algorithm.
The maximum flow problem. Minimum cut. Max-flow min-cut theorem (5 hours). Ford-Fulkerson algorithm. Residual networks. Edmonds-Karp algorithm
Minimum cost flow problem (3 hours). The general character of the problem. Negative cycle

optimality conditions and corresponding algorithm.

Probability and Statistics
Trials, events, probability. Properties. Examples. **(2 hours)**
Conditional probability, independence. **(2 hours)**
Full probability formula. Bayes Theorem. Examples. **(2 hours)**
Discrete random variables. Distribution, mathematical expectation, dispersion. **(3 hours)**
Expectation and dispersion properties. **(2 hours)**
Continuous random variables. Distribution, density. Examples. **(3 hours)**
Mathematical expectation and dispersion. Properties. Examples **(2 hours)**
Random variables functions. Distribution. Mathematical expectation. **(2 hours)**
Normal distribution. **(2 hours)**
Discrete random vectors. Distribution. Covariance. **(2 hours)**
Law of large numbers. Some applications with normal distribution. **(2 hours)**
Descriptive statistics. Variables. Classification. Measures of location. **(2 hours)**
Measures of variability. Numerical and graphical description. **(2 hours)**
Random selection. Point estimation and their properties. **(2 hours)**

Skills given

Gains basic knowledge about graphs, their structure and properties. Is fully acquainted with the problems formulated on graphs and networks, their applications in practice, as well as with algorithms for their solution. This subject aims to develop the basic elements of probability and statistics, familiarity with the basic concepts and techniques that accompany their use. Its content is of the basic formation type in mathematical statistics.

Literature

Base literature:

1. **Llukan Puka.** Probabilitetet dhe Statistika e Zbatuar: koncepte themelore. Shtepia Botuese e Librit Shkollor (e re), Tirane, 2008.
2. **Kedhi V.** Teoria e Grafeve dhe rrjedhat në Rrjeta, shblu, 2000.

Recommended literature:

1. **Douglas B. West.** Introduction to Graph Theory, Prentice-Hall, 2006.
2. **JayYellen, Jonathan L. Gross**. Graph Theory and Its Applications (Discrete Mathematical & Applications Series), CRC, 1998.
3. **Joan M. Aldous, Robin J. Wilson**. Graphs and Applications, An introductory Approach, Springer, 2000.
4. **Ravindra K. Ahuja, Thomas L. Manganti, James B. Orlin.** Network flows, Theory, Algorithms and Applications. Prentice-Hall, 1994.

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Java Programming

Teaching Activities	Lectures	Exercises	**Lab. Work**	**Total**
Frequentation Requirement	Non mandatory	75%	100%	0
Teaching Hours	60	60	15	135
Individual Study Hours	180			
Language of Instruction	Albanian			
ECTS Credits	12			
Teaching Methods	Second Year, First Semester, 15 weeks: 2 hours of lecture, 2 hours of exercise / week Second Year, Second Semester, 15 weeks: 2 hours of lecture, 2 hours of exercise, 1 hour of lab. Work / week			
Scoring Details	Written exam and assigments in laboratory			
Assessment	Laboratory	10%		
	Final exam	90%		
	Total	**100%**		
Main Instructor	**Prof.Asoc. Dr. Alda Kika,** *e-mail:alda.kika@fshn.edu.al*			

Course Description	This course aim is to provide the students with ability to write programs in Java language. It is oriented towards the theoretical acquisition of the language and development of practical skills including object orienting programming, data structures in java and building graphical applications or applets that can be used in internet. The course includes teaching the syntax of the Java language, characteristics of the language, compilation and execution of a program, building classes and testing them. During this course students will learn the elements needed to build a complex application with user interface which include the use of components to get the data of the user and display results through the components, handling the events, reading and writing files through the Java language classes, using default classes in Java or creating them for data structures and their implementation in applications.
Objectives	The objectives of this course aims that at the end of this course students will be able to: understand the art of programming and particularly the structure and syntax of the Java programs. build, edit, compile, test and execute programs and applets in Java. build and implement simple applications with user interface. understand and use interfaces and internal classes. understand and use events handling in Java using the Java classes of events. understand the different components of the Swing package and know when and how to use them in applications with user interface. understand and use exceptions in Java. understand the files and Java classes needed to implement read and write operations. become familiar with simple and advanced data structures and use them in an application with Java language. build a complex application with many classes.
Preliminary Requirements	Basics of Informatics, Data Structures in C language.
Lecture Topics	
Java Programming Language introduction.	

Characteristics, advantages of java. Short history of java development and the technology necessities for creating java language. Differences with other languages (similarities and differences). Platform independence. Description of security elements. **(2 hours)**

The structure of a simple program in Java.

Describe the building blocks of a simple program and its characteristics. How to display special characters. Fundamental data types. Using default classes in programs. **(2 hours)**

Variables and constants.

Assignment and initialization. Casting. Math operators and statics methods of Math class. Variables and their declaration. Arithmetic expression using Arithmetic Operators and statics methods of Math class. **(2 hours)**

The String Class.

String manipulation. Reading input from the user.

Declaring and using string variables. Use methods in String class to manipulate character strings. Using Scanner class for reading input from the user. **(2 hours)**

Implementing classes.

Classes introduction: definition and usages. Construction of the class. Variables and methods. Instance variables declaration for implementing attributes in a class. Using methods for implementing its behavior. **(2 hours)**

Classes Constructors.

Constructing constructors. Differences between primitive type and reference type. Constructors: usage and their construction. Example on using classes in applications. Reference copying. Null References. **(2 hours)**

The If Statements.

Relational operators. Comparing fundamental data types, strings and objects. Implementing decision making using the if statement. Examples using if, if else and if statements with multiple conditions. Using relational operators for comparing values of fundamental types. **(2 hours)**

Multiples alternatives.

If else, if statements and switch statement. Using Boolean expression. The Boolean operators. If else if and switch statements for multiple conditions. Boolean type. Building and using Boolean expression using Boolean operators. De Morgan's Law. **(2 hours)**

Iteration using while and for loops.

Use nested loops to implement multiple levels of iterations. Manipulating data for users. Using Random Class for random numbers simulations. **(2 hours)**

Array and ArrayList.

Construct arrays, assignment, initialization and access the stored values. Declaration and manipulation of two-dimensional arrays. Implement array and ArrayList algorithms. **(2 hours)**

Applets. Implementing applets and running them using an HTML file.

Applet introduction. What is an applet, where to use it and differences between applets and applications.

HTML files introduction and putting an applet in a HTML page. Implementing a simple applet. Applet cycle. **(2 hours)**

Graphical elements: ellipse, rectangle, lines.

Fonts and colors. Displaying ellipse, rectangle and lines in applet. Learning predefined colors. Displaying text with different fonts. Using dialog boxes for input and output. Reading input. Introduction to methods of the applet cycle. **(2 hours)**

Designing Classes.

Discovering classes and their relationships. The concepts of cohesion and coupling. UML notation for displaying class relations. Immutable classes. Minimize the use of effects. Statics methods. Statics variables. Scope rules for local variables and instance variables Packages. **(2 hours)**

Inheritance.

Subclasses and superclasses construction and their convertion. Implementing subclasses and superclasses. Reasons for using inheritance and its characteristics. Reciprocal convertion between subclasses and superclasses references. Inheritance of attributes and behaviour. **(2 hours)**

Interfaces and Polymorphism.

Interface: a mechanism for supporting multiple inheritance in Java. Implementing an interface and its characteristics. Abstract classes. Differences between abstract classes and interfaces. Access levels in

Java and their differences. Object Superclass and overriding of its most frequently used methods. **(2 hours)**

Principes of designing user graphical interface.

The model and hierarchy of events in Java. Graphical applications in Java. Differences between them and keyboard applications or applets. Event model. Events, Event Sources, and Event Listeners. Event Adapters. **(2 hours)**

Mouse Events. Frames.

Methods to capture mouse events. Implementing event listeners as inner classes. Implementing applets that process mouse events that interact with the user. The structure of a frame. **(2 hours)**

Graphical applications with frames.

How to display information in a frame. Opening and closing a frame. Event actions. Building graphical applications with buttons. Processing Timer Events. **(2 hours)**

Graphical User Interfaces Management.

Layout Management: BorderLayout, FlowLayout, GridLayout. Adding components to a container. Processing user input through text area in a graphics application. **(2 hours)**

Choices.

Example of an application that uses choice components.
Building and using menus and exploring the Swing Documentation. The elements of the menu structure. Examples of using menus. JSlider and JScrollPane components. **(2 hours)**

Exception handling in Java.

Exception in java.
Reasons of using exceptions.
Types of exceptions. Implementing and catching exception in java.
Examples of using exceptions. **(2 hours)**

Disk Files. Streams of bytes and characters

Hierarchy of classes for reading and writing the streams of bytes and characters.
Reading, writing and accessing sequences of characters.
Java classes for reading/writing a stream of bytes and characters.
Differences between text files and binary files.
Reading, writing and processing text files. Examples.
Row commands arguments. Using row commands arguments in applications. **(2 hours)**

Random access to the data in the file.

Creating, writing and processing data in a random access file. **(2 hours)**

Object Streams to automatically read and write entire objects.

Reading and writing objects using ObjectOutputStream and ObjectInputStream. **(2 hours)**

Multimedia in applet and applications.

Uploading and displaying an image in applet.
Uploading and playing music files. **(2 hours)**

Recursion.

The technique of recursion. Examples. The relationship between recursion and iteration. Analyze problems that are much easier to solve by recursion than by iteration. Using recursive helper methods. **(2 hours)**

Introduction to Data Structures.

Using and implementing linked lists and stacks. Add and remove elements efficiently in a list. Stack and queue data types. **(2 hours)**

Advanced data structures.

Sets and maps. Hash Tables. Binary Search Trees. Describe preorder, inorder and postorder traversal in binary search trees. Priority Queues. Heaps. **(2 hours)**

Generic Programming.

Generic classes and type parameters. Generic Methods. **(2 hours)**

Laboratory Topics

Implementation of user interface applications that use components and menage events.**(6 hours)**
Implementation of applications with input and output actions in text files, random access files and object files. **(5 hours)**
Usage of advanced structures in applications.**(4 hours)**

Developed Skills

At the end of the course, students should be able to identify and develop concepts of object-oriented programming such as inheritance and polymorphism using Java language, draw graphics in Java, build applications with user interface, select and use classes and other packages in Java, and have the ability to program independently to solve real life problems.

Literature

Basic Text Book:

Cay Horstmann: Big Java, 4th edition, John Wiley 2010

Reccomended Literature:

Giovanni Pighizzini, Mauro Ferrari:Dai fondamenti agli oggetti - Corso di Programmazione Java Terza Edizione, Pearson Education, 2008

Pellegrino Principe: Java 7, Apogeo 2011.

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Multimedia Systems

Teaching Activities	Lectures	Exercises	Lab. Work	Total
Frequentation Requirement	Non mandatory	75%	100%	0
Teaching Hours	30	30	0	60
Individual Study Hours	115			
Language of Instruction	Albanian			
ECTS Credits	7			
Teaching Methods	Second Year, Second Semester, 15 weeks: 2 lecture hours, 2 exercise hours			
Scoring Details				
Assessment	Semestral project	5%		
	Participation	5%		
	Final exam	90%		
	Total	**100%**		
Main Instructor	**Prof.Assoc.Dr. Ana Ktona**			

Course Description	This course provides an introduction to Multimedia Data, Multimedia Applications and Multimedia Systems. World Wide Web and its technologies will be used to treat practically the concepts.
Objectives	The goal of this course is to help students become familiar with the Multimedia Data, Multimedia Applications and Multimedia Systems.
Preliminary Requirements	Basic concepts in algorithms and data structures.

Lecture Topics

1. Introduction to Multimedia Data, Multimedia Applications and Multimedia Systems. (2 hours)
2. Multimedia Data Representations. (2 hours)
3. Multimedia and web technologies - HTML. (2 hours)
4. Multimedia and web technologies - HTML5. (2 hours)
5. Multimedia and web technologies - Cascading Style Sheets ™: Basic Knowledge,. (2 hours)
6. Multimedia and web technologies - Cascading Style Sheets ™ CSS Layout (2 hours)
7. Multimedia and web technologies - Cascading Style Sheets ™: Transforms, Transitions. (2 hours)
8. Multimedia and web technologies - Cascading Style Sheets ™, Animations, User Interface. (2 hours)
9. Multimedia and web technologies - JavaScript: Introduction to Scripting. (2 hours)
10. Multimedia and web technologies - JavaScript: Control Structures. (2 hours)
11. Multimedia and web technologies - JavaScript: Functions. (2 hours)
12. Multimedia and web technologies - JavaScript: Arrays. (2 hours)
13. Multimedia and web technologies - JavaScript: Objects. (2 hours)
14. Multimedia and web technologies – JavaScript: Document Object Model. (4 hours)

Developed Skills

Upon completion of this course, students should be able to understand the fundamental concepts in Multimedia and create Web Based Multimedia Applications.

Literature

Xhina Endrit; Ktona, Ana. Sisteme Multimediale, InfBotues 2010
Ktona, Ana; Xhina Endrit. Sisteme Multimediale Ushtrime, InfBotues 2013

C++ Programming

Teaching Activities	Lectures	Exercises	Lab. Work	Total
Frequentation Requirement	Non mandatory	75%	0%	0%
Teaching Hours	30	30	0	60
Individual Study Hours	100			
Language of Instruction	Albanian			
ECTS Credits	**6**			
Teaching Methods	Second Year, First Semester, 15 weeks: 2 lecture hours, 2 exercise hours,			
Grading System	Written exam			
Assessment	Active participation			
	Periodic assessment			
	Project			
	Labs			
	Internships			
	Final exam	100%		
	Total	**100%**		
Main Instructor	**Xheni MELO**			

Course Description	The course introduces the main concepts of Object Oriented programming, focusing briefly on some basic aspects of procedural programming. Basic concepts of object-oriented programming are introduced, such as classes, objects, inheritance, data abstraction, encapsulation, polymorphism. Exception handling techniques are also introduced, as well as testing an application.
Objectives	The objective of the set of C++ lectures is to provide the students the basic principles and knowledge of C++ programming language. It covers structured programming and object oriented programming in C++, having as a final goal the development of basic programming and problem solving skills for the students. The set of algorithms lectures aims the development of advanced skills in solving problems computationally, development of analytical skills and of advanced programming skills. These lectures aim to orient the students toward efficient solutions for known computational problems.
Preliminary Requirements	**Introduction to Informatics.** **Data structure.**
Lecture Topics	
1. Basic programming constructs 1. Conditional structures. Cyclic Structures (2 hours) 2. Basic programming constructs 2. Functions. Components of a function and its types. Value and reference parameters. Functions in relation to tables and structures. Overload functions.	

Recursive functions (2 hours)

3. Introduction to object-oriented programming. First introduction to the basic concepts of object-oriented programming (2 hours)

4. Classes and facilities. The class and its members. Constructors, attributes, methods, static members. Accessibility of class members. Using the classroom and creating objects (3 hours)

5. Class inheritance. Types of heritage. Constructors in inheritance. Multiple Inheritance (4 hours)

6. Composed classes. Classes consisting of objects of other classes. Constructor in composition. Use of composed classes (2 hours)

7. Abstraction and Encapsulation. Introduction to the concepts of data abstraction and encapsulation, as two of the main concepts of object-oriented programming. Hiding data / functionalities and packing them into a single entity (3 hours)

8. Polymorphism. Introduction to the concept of polymorphism, its types and concrete uses. (2 hours)

9. Overloading and overriding Reimplementing the functionality of a class (2 hours)

10. Treatment of exceptions. Mechanisms for handling exemptions. Techniques of their use. Structure try..catch. (2 hours)

11. Overload operators. Introduction to the concept of creating overload operators to be used with objects of a class. (2 hours)

12. Input / Output operations and files Reading from the keyboard and writing on the screen. Reading and writing to file. Binary files (2 hours)

13. Testing Introduction to the testing techniques of an application (2 hours)

Tutorial Topics

1. Use of IF, IF..ELSE control structures. FOR, WHILE and DO..WHILE cycles (2 hours)

2. Creation and use of functions. Using value and reference parameters and illustrating different types of functions (2 hours)

3. The first presentation with the basic concepts of object-oriented programming (2 hours)

4. Creating classes and using objects.Classes and their members. Constructors, attributes, methods, static members. Accessibility of class members. (3 hours)

5. Illustration of the concept of inheritance. Types of heritage. Constructors in inheritance. Multiple Inheritance (4 hours)

6. Illustration of the concept of composition. Classes consisting of objects of other classes. Constructor in composition. Use of composed classes (2 hours)

7. Introduction to the concepts of data abstraction and encapsulation, as two of the main concepts of object-oriented programming. Hiding data / functionalities and packing them into a single entity

(3 hours)

8. Introduction to the concept of polymorphism, its types and concrete uses. (2 hours)

9. Reimplementation of the functionalities of a class with the concept of overloading and overriding (2 hours)

10. Illustration of mechanisms for handling exemptions. Techniques of their use. Structure try..catch. (2 hours)

11. Creating overload operators to be used with objects of a class. (2 hours)

12. Perform Input / Output operations and files Read from the keyboard and write on the screen. Reading and writing to file. Binary files (2 hours)

13. Introduction to the testing techniques of an application (2 hours)

Labs Topics

1. Introduction to C++ programming. Basic elements of a C++ program. Data types, operators, statements. (2 hours)

2. Control structures. Relational and logical operators. Selection control structures, If. Else and Switch. Looping control structures. (2 hours)

3. User defined functions. Types of user defined structures. Actual and formal parameters. Value and reference parameters. Scope of identifiers. (2 hours)

4. Input / Output. Usage of *cin* and *cout*. Usage of functions to format input/output streams. File input /output. (2 hours)

5. Arrays and Strings in C++. Usage of strings and arrays in C++. String functions. (2 hours)

6. Vector Type and Records. Introduction to Vector Type and its functions. Comparison with Arrays. Introduction to records and their usage in functions. Usage of records with arrays. (2 hours)

7. Classes and data abstraction. Introduction to Object Oriented Programming. Structure of a class and its usage. Members of a class. Information hiding and abstraction (2 hours)

8. Class Inheritance. Types of inheritance. Method overloading and overriding. Constructors of a derived class. (2 hours)

9. Composition. Characteristics of a composed class. Constructors. Static members of a class. (2 hours)

10. Exception handling. Handling Exceptions within a Program. Try catch block. Using C++ Exception Classes (2 hours)

11. Overload operators. Overloading binary and unary operators. Friend functions with a class. Overload functions with friend functions and member functions. (2 hours)

12. Templates. Function templates. Class templates. Usage of templates (2 hours)

13. Pointers. Pointer variables declaration. Pointer usage. Dynamic variables. Dynamic arrays (2 hours)

14. Linked lists. Linked lists classes (2 hours).

15. Practical implementation of algorithms in C++, measuring of the algorithm execution time for the best, average and worst case. (1 hour)

16. Practical implementation of algorithms in C++. Measuring of algorithms execution time and identification of the its dependency from the size and type of input data. (2 hours)

17. Practical implementation in C++ of stacks, queues and related algorithms. (1 hour)
18. Practical implementation in C++ of algorithms with hash tables. (1 hour)

19. Practical implementation in C++ of algorithms with binary trees and binary search trees. (4 hours)
20. Practical implementation in C++ of graphs and related. (4 hours)

21. Practical implementation in C++ of searching and sorting algorithms and comparison of their execution time. (2 hours)

Developed Skills

1. Good theoretical and practical knowledge of basic concepts of object-oriented programming.
2. Using Object Oriented concepts and the opportunities they offer, in problem solving
Build the most efficient programs that offer the best solution to problems.

Literature

Textbooks:
"C++ dhe programimi i orientuar në objekte" – **Dr. Xheni MELO**

FACULTY OF NATURAL SCIENCES
DEPARTMENT OF INFORMATICS

Algorithms

Teaching Activities	Lectures	Exercises	Lab. Work	Total
Frequentation Requirement	Non mandatory	75%		0%
Teaching Hours	30	30	0	60
Individual Study Hours				
Language of Instruction	Albanian			
ECTS Credits	6			
Teaching Methods	Second Year, Second Semester 15 weeks: 2 lecture hours, 2 exercise hours, 1 lab hour.			
Grading System	Written exam			
Assessment	Active participation			
	Periodic assessment			
	Project			
	Labs			
	Internships			
	Final exam	100%		
	Total	**100%**		
Main Instructor	**Bora BIMBARI**			

Course Description	This course consists of lectures and tutorials that introduce the student with well known solutions developed for many standard computational problems.
Objectives	The set of algorithms lectures aims the development of advanced skills in solving problems computationally, development of analytical skills and of advanced programming skills. These lectures aim to orient the students toward efficient solutions for known computational problems.
Preliminary Requirements	**Introduction to Computer Science** **Data structures**

Lecture Topics

1. **Introduction to algorithms (2 hours).** Definition of the algorithm. Algorithms properties: correctness, efficiency, simplicity. Algorithms running time for the best, worse and average case.

2. **Proving algorithms correctness (1 hour).**
3. **Asimptotic growth, algorithms complexity (1 hour).** Dependancy of algorithms' execution time from the input data (data size and data type). Formal expression of the execution time dependancy from the data via mathematical notations O, Ω, Θ.

4. **Abstract data type. Relationship between the algorithm and the type of the data it manipulates (2 hours).** Introduction to the abstract data type concept. Difference between data types and data structures. Stacks, queues, priority queues, linked lists, dynamic tables and related algorithms.

5. **Hash tables. Algorithms for hash tables. (2 hours).** Dictionary data type. Direct access tables. Usage of hash functions to define the position of the data in tables based on their key values. Using hash tables. Dealing with collisions: chaining method and open addressing. Algorithms for inserting, searching, and removing elements from the hash table.

6. **Binary search trees. Algorithms for solving problems using binary search trees. (4 hours).** Definition of the binary tree. Definition of the binary search tree. Inserting, deleting, and searching elements in the binary search tree. Balanced trees. Inserting, deleting, and searching elements in balanced trees.

7. **Graphs and related algorithms.Grafet. (6 hours).** Defining data structures for representation of graphs: adjacency matrix, adjacecny lists. Breadth-first-traversal algorithm. Depth-first traversal algorithm. Finding the shortest path in the graph, finding cycles in the graph, finding connected components in the graph. Bellman-Ford Algorithm. Dijkstra Algorithm. Floyd-Marshall algorithm.

8. **Divide and concquer technique for problem solving. (2 hours).** Dividing the main problem into identical subproblems, integration of the solution of each subproblem to obtain the main problem solution. Mergesort, quicksort, and binarysearch algorithms.

9. **Dynamic Programming (2 hours).**

10. **Greedy algorithms (2 hours).**

11. **String comparison (2 hours).** Searching for common subsequences in strings.

12. **Indexing and searching in the world wide web (2 hours).** Building indexes. Searching using indexes. Principles of searching in the world wide web.

13. **Sorting and data retrieval from the world wide web (2 hours).** *PageRank* used by Google for searching in the web.

Tutorial Topics

1. Definition of the algorithm. Algorithms properties: correctness, efficiency, simplicity. Algorithms running time for the best, worse and average case. (2 hours)

2. Exercises on proving the correctness of algorithms (1 hour).

3. Dependency of algorithms' execution time from the input data (data size and data type). Formal expression of the execution time dependency from the data via mathematical notations O, Ω, Θ. (1 hour)

4. Abstract data type concept. Difference between data types and data structures. Stacks, queues, priority queues, linked lists, dynamic tables and related algorithms. (2 hours)

5. Dictionary data type. Direct access tables. Usage of hash functions to define the position of the data in tables based on their key values. Using hash tables. Dealing with collisions: chaining method and open addressing. Algorithms for inserting, searching, and removing elements from the hash table (2 hours)

6. Exercises on the definition of the binary tree, definition of the binary search tree. Inserting, deleting, and searching elements in the binary search tree. Balanced trees. Inserting, deleting, and searching elements in balanced trees (4 hours)

7. Defining data structures for representation of graphs: adjacency matrix, adjacency lists. Breadth-first-traversal algorithm. Depth-first traversal algorithm. Finding the shortest path in the graph, finding cycles in the graph, finding connected components in the graph. Bellman-Ford Algorithm. Dijkstra Algorithm. Floyd-Marshall algorithm. (6 hours)

8. Exercises for Divide and conquer technique, algorithms merge sort, quicksort, binary search. (2 hours)

9. Exercises on dynamic programming (2 hours).

10. Exercises on 'greedy' algorithms (2 hours).

11. Searching for common subsequences in strings. (2 hours)

12. Building indexes, searching through indexes. Searching principles in the world wide web. (2 hours)

13. Exercises on the PageRank system, used by Google for the search in the world wide web. (2 hours).

Developed Skills

1. Implementation of algorithms in C++.
2. Undersanding of problems and their specification.
3. Identification of appropriate data structures for representing the data related to the problems introduced.
4. Development of correct algorithms for specified problems.
5. Identificatio of the best, worst and average case of the algorithm execution time and calculation of the complexity.

Literature

Textbooks:

D.S.MALIK; C++ Programming: From Problem Analysis to Program Design.
Cormen T, Leiserson C, Rivest R, Stein C. Introduction to Algorithms, third edition.
Skiena Steven. The Algorithm Design Manual, second edition.

Lecture Notes:

BIMBARI Bora. Lecture in Algorithms.

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Computer Organization And Data Structure

Teaching Activities	Lectures	Exercises	Lab. Work	Practice	Total
Frequentation Requirement	Non mandatory	75%	0%	0%	
Teaching Hours	30	15	0	0	**45**
Individual Study Hours	105				
Language of Instruction	Albanian				
Evaluation method	Exam				
ECTS Credits	6				
Teaching Methods	Second Year, Third Semester, 15 weeks: 2 lecture hours, 1exercise hour				
Scoring Details	Written exam, 45-50 points for a grade of five, every ten additional pointsgrading is increased by one.				
Assessment, Note , Exam	Semestral project	0%			
	Participation	0%			
	Laboratory	0%			
	Project	0%			
	Practice	0%			
	Final exam	100%			
	Total	**100%**			
Main Instructor	**Agim Çami**,e-mail: agim.cami@fshn.edu.al **Julian Fejzaj**,e-mail: julian.fejzaj@fshn.edu.al				

Course Description	The course deals with all the construction and operation of the computer. This course offers important concepts such as computer number formats, mathematical and logical operators, structure of instructions, ISA, CPU functions, micro machines, "pipeline" processors, sequencers, memory hierarchy, characteristics of RAM memories, Cache memory, Interfacing CPU and Peripherals, interruptions, DMA, I/O processors, etc.
Objectives	Study and analysis of the construction, operation and factors that affect software and hardware components of a computer system.
Preliminary Requirements	Data Structure

Lecture Topics

Part I. The units of a BASIC computer (4 hrs)
Function and structure of control unit
Memory, the memory hierarchy
Data format and the processing unit
Adders, Synthesizing the ALU

Part II. The Directions and ways of addressing (5 hrs)
VirtualMachines
Structure of instruction , theoretical structure , Real structure instructions. Examples of instruction formats.
Machines with zero,one, two and three addresses
Ways of addressing , logical, physical , effectiveaddress, etc…
Classification of instructions
RISC and CISC processors, comparison

Part III. The entity CENTRAL ORGANIZATION (4 hrs)
Ways of communication within the central unit , busses.

constructing a datapath , single-cycle datapath, multicycledatapath
Structures withone,two, three busses
Organization of the central unit with general registers
Pipeline processors , the implementation of pipeline , pipeline hazards , etc…

Part IV. Control Unit . Sequencer(4 hrs)
Definitions ,clock concept
Hardware sequencers
Micro programmed sequencer, Wilkes's model, addressing of the microinstructions
Analysis of an elementary processor
Performances of processors and their measurement

Part V. Memory Organization (4 hrs)
Technologies and the hierarchy of memory
Characteristics of memory devices
Methods of capturing information (Access Modes)
SRAMMemories, DRAM Memories
The cache memory, operating parameters, types etc.

PART VI. Communication of the central sub-system with external environment (5 hrs)
Function and structure of the peripheral sub-system
Interfaces, busses I/O
Ways or techniques of achievingI/O
Delay execution technique, examples
Status , pooling technique, examples
Interrupt technique, Mode interruption, interrupt disabling,masking,priorityetc.
DMA, DMA controller construction, operation, DMA modes
I/O processor, I / O instructions, channel programs, Organization of I/O Proccessors
PART VII. Parallel Processor (4 hours)
Parallel processor architecture (SIMD, MIMD, MISD, VLIW, etc.). Their specifications.
Typical organization of a SIMD processor. Possible extensions of a SIMD. Examples.
MIMD architectures. Construction and operation of a MIMD processor.
Construction of a VLIW processor. Explicit parallelism (EPIC).

Developed Skills

Exercises and case studies about **: Computer Organization (2 hours)**
Exercises and case studies about **:Computer Arithmetic. (2 hours)**
Exercises and case studies about **:Instruction Set Architecture. (3 hours)**
Exercises and case studies about **:Central Processing Unit. (3 hours)**
Exercises and case studies about **:Computer Performance Evaluation. (3 hours)**
Exercises and case studies about **:Computer Memory Subsystem. (2 hours)**

Skills and knowledge gives to the student

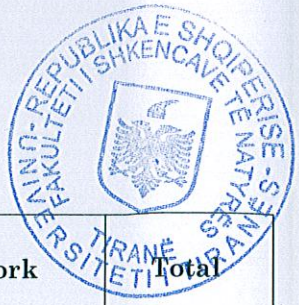
Students at the end of this class will be able to :
Understand the change of the instructions .
Understand the micro-architecture of the processors and pipeline technology.
Understand the organization of memory, virtual memory and cache.
Understand the function of the mechanism of entry-exit.
Architecture of parallel processors

Literature
Agim Çami
Computer architecture, first edition, 2011

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Agim Çami
Computer Organization and Architecture - Through examples and exercises 2012

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS



UML

Teaching Activities	Lectures	Exercises	Lab. Work	Total
Frequentation Requirement	Non mandatory	75%	100%	0
Teaching Hours	30	30	0	60
Individual Study Hours	90			
Language of Instruction	Albanian			
ECTS Credits	**6**			
Teaching Methods	Second Year, Second Semester, 15 weeks: 2 lecture hours, 2 exercise hours			
Scoring Details	Continuous assessment/final exam, 45-50 points for a grade of five, every ten additional points grading is increased by one.			
Assessment	Semestral project	40%		
	Final exam	60%		
	Total	**100%**		
Main Instructor	**M.Sc. Klesti Hoxha,** *e-mail: klesti.hoxha@fshn.edu.al*			

Course Description	This course gives basic knowledge of object oriented modeling using UML. It describes use case diagrams, class diagrams, object diagrams, sequence diagrams, etc. It includes a semestral project which should provide the design of an innovative software solution applicable in business or society using a model-driven process.
Objectives	1. Introduction to software modeling activities. 2. Introduction to the main diagrams of UML. 3. Improvement of knowledge in object oriented programming concepts.
Preliminary Requirements	Students should have fundamentals skills in an object oriented language. (i.e. JAVA, C++)

Lecture Topics

1. **Introduction to modeling.** Initial presentation of UML as a modeling language and its uses. **(2 hours)**
2. **Object Oriented Analysis and Design.** The description of the principal stages in object oriented modeling, analysis/design. UML diagram types. **(2 hours)**
3. **UML modeling helping tools. (2 hours)**
4. **Use Case diagrams. (4 hours)**
5. **Basic concepts of class diagrams. (4 hours)**
6. **Advanced class diagrams. (2 hours)**
7. **Case study:** The modeling of a software system using use case, class and object diagrams. **(2 hours)**
8. **Object diagrams. (2 hours)**
9. **Sequence diagrams. (4 hours)**
10. **Activity diagrams. (4 hours)**
11. **Case study: The design of a persistence framework. (2 hours)**

Developed Skills

1. Conceptual skills in object oriented modeling.
2. Basic knowledge of the Unified Modeling Language.
3. Using UML as a helping tool in developing software systems.
4. The improvement of effective team work skills while collaborating for the semestral project.

Literature

Required texts:

R. Miles and K. Hamilton. Learning UML 2.0, O'Reilly 2006

Martin Fowler . UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition),

Addison-Wesley 2004

Recommended Reading List:

Applying UML and Patterns (3rd edition), Craig Larman, Addison-Wesley 2004

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Third Year Courses

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Operating Systems

Teaching Activities	Lectures	Exercises	**Lab. Work**	**Total**
Frequentation Requirement	Non mandatory	75%	100%	0
Teaching Hours	29	29	0	58
Individual Study Hours	90			
Language of Instruction	Albanian			
ECTS Credits	10			
Teaching Methods	Third Year, First and Second Semester, 29 weeks: 2 lecture hours, 2 exercise hours			
Scoring Details	The minimum number of points in order to pass is 40.			
Assessment	Class participation	5%		
	Projects	15%		
	Final exam	80%		
	Total	**100%**		
Main Instructor	**Msc. Denada Xhaja,** *e-mail:denada.xhaja@fshn.edu.al*			

Course Description	This course introduces the basic facilities provided in modern operating systems as well as principles of distributed systems. It is devided into two major sections. The first section discuses the evolution of operating systems, the main components of OSs' such as process management, memory management, I/O management, file system management, etc. The second section covers principless and issues related to distributed systems.
Objectives	The purpose of this course is providing a clear: understanding of the concepts that underlie operating systems. understanding of the concepts that underlie distributed systems.
Preliminary Requirements	Data structures Computer system organization C or Java programming
Lecture Topics	
Introduction to OS and the history of OS evolution. **Computer system organisation.** **Operating system structure.** **Preocesses and threads.** **CPU scheduling.** **Process synchronization** **Deadlocks.** **Main memory.** **Virtual memory.** **Massive Storage.** **File system.** **I/O system** **Protection and security.** **Introduction to Distributed Systems.** **Communication in DS.** **Processes in DS.** **Naming in DS.** **Synchronization in DS.** **Consistence and replication.** **Fault tolerance.**	
Developed Skills	

Upon sucessful completition of this course, students would be able to
1. understand the role of Operating Systems
2. Apply the concept of a process, thread and scheduling algorithms.
3. Apply the concepts of process synchronization and how it is achieved as well as realize the concept of deadlock and different ways to handle it.
4. Realize various memory management techniques.
5. Realize the concept of I/O management and File system
6. Understand the core concepts of distributed systems such as access and location transparency, parallelization of tasks, Fault-tolerance, etc.

Literature

Peter B. Galvin & Silberschatz, **Operating system concepts 8th ed.**
Andrew S. Tanenbaum & Maarten Van Steen, Distributed Systems: Principles and Paradigms, 2nd Edition
Williams Stallings, **Operating systems Internals and Design principles, 7th Edition**
A. Tannenbaum & A. Woodhull, **Operating Systems Design And Implementation, 2nd Edition**
George Coulouris, Jean Dollimore, and Tim Kndberg, Distributed Systems: Concepts and Design, 5rd edition

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Networks

Teaching Activities	Lectures	Exercises	Lab. Work	Practice	Total
Frequentation Requirement	Non mandatory	75%	0%	0%	
Teaching Hours	58	58	0	0	116
Individual Study Hours	109				
Language of Instruction	Albanian				
Evaluation method	Exam				
ECTS Credits	9				
Teaching Methods	Third Year, Fifth Semester, 30weeks: 2 lecture hours, 2exercise hour				
Scoring Details	Written exam, 45-50 points for a grade of five, every ten additional pointsgrading is increased by one.				
Assessment, Note , Exam	Semestral project	0%			
	Participation	0%			
	Laboratory	0%			
	Project	10%			
	Practice	0%			
	Final exam	90%			
	Total	100%			
Main Instructor	Julian Fejzaj,e-mail: julian.fejzaj@fshn.edu.al				

Course Description	This module presents the architecture, structure and network layers, main network protocols and IP protocol addressing scheme The course deal with important network directory services, different techonologies and methods related with network security and network performance. Network course also describes routing protocols ilustrated with configuration of router Internetwork Operating System(IOS)
Objectives	The course aims to enable students to connect network devices using different network media types to configure networks refer to IP addressing and subnetting installing and troubleshooting active directory and router configuration
Preliminary Requirements	Computer Organization and Architecture

Lecture Topics

Introduction to Networking, what is computer networking, network topologies,wireless topologies,network classifications Lan, Man, Wan.Network modelsclient/server and peer-to-peer networking (3 hrs)
Media and connectors,Networking media, cables coaxial, twisted-pair, fiber. Media connectors, *Media Characteristics*. Networking components and devices, specialized network devices.(4 hrs)
Ethernet Networking Standarts, ethernet standarts,characteristics specified in the IEEE 802.x , wireless standardsspecified in 802.11 standard. (3 hrs)
OSI Model, layers of the OSI model, the layers at which devices operate. (2 hrs)
Network Protocols,routed protocols, non- routed protocols, IPX/SPX protocol, TCP/IP suite protocols,comparing OSI and TCP/IP model ,important protocols of TCP/IP. (6 hrs)
IP protocol, TCP/IP Addressing scheme,public and private addresses, TCP/IP ports and protocols,network address classes, TCP/IP Addressing scheme for Ipv4,Ipv6.IP Addressing and subnetting,subnetting using variable length subnet mask (VLSM). (5 hrs)

Wide Area Networking and Internet Access, Switching methods, technologies used for accessing the internet, x-DSL, ISDN, remote access protocols. (3 hrs)
Virtual Private Networks VPN, Introduction to VPN, VPN protocols, VPN security, types of VPN, configuration of VPN network. (2 hrs)
Fault tolerance,Network Performance and Optimization, RAID levels, the cluster concept, disaster recovery, data continuity strategies, remote sites. (3 hrs)
Network monitoring and troubleshooting, Identify symptoms and problems using ICMP, SNMP protocols. Physical issues, logical issues, TCP/IP configuration and command-line utilities. (3 hrs)
Wireless Networking, Characteristics of wireless communication standards, implementing a wireless network, securing wireless networking. (2 hrs)
Network Security, inside network security, the efect of viruses in networks, outside attacks,physical network security,logical network security , firewalls , proxy servers , Authentication methods, understanding network security threats (4 hrs)
Routing,Elements of a router, basic router configuration, internetwork operating (IOS) distance-vector routing protocols, link-state routing protocols.Configuration of RIP, IGRP, OSPF routing protocols. Interior Routing Protocols, Border Gateway protocol. (6 hrs)
Active Directory Services (12 hrs)
Mail service, DNS service, DNS zones, DHCP service.
Definition of domain, Domain Controller, active directory tree and forest structure
Configuring sites ,configuring replication and active directory optimization
Organization units, active directory security policies,group policy objects(GPO)and their configuration.
Backup and recovery of active directory, restoringactive direcory

Seminar Topics

Introduction to Networking, network topologies,network classifications Lan, Man, Wan.Network modelsclient/server and peer-to-peer networking (3 hrs)
Media and connectors, Networking media, cables coaxial, twisted-pair, fiber. Media connectors, *Media Characteristics*. Networking components and devices, specialized network devices.(4 hrs)
Different Networking Standarts, exercices withIEEE 802.x, 802.11 standarts (3 hrs)
OSI Model,discussion questions examples about layers of the OSI model, the layers at which network devices operate. (2 hrs)
Network Protocols,routed protocols, non- routed protocols, IPX/SPX protocol , TCP/IP suite protocols,comparing OSI and TCP/IP model ,important protocols of TCP/IP. (6 hrs)
IP protocol, TCP/IP Addressing scheme for Ipv4,Ipv6. IP Addressing and subnetting exercices ,subnetting exercices using fixed length subnet mask(FLSM), variable length subnet mask (VLSM). (5 hrs)
Wide Area Networking and Internet Access, Switching methods, technologies used for accessing the internet, x-DSL, ISDN, remote access protocols. (3 hrs)
Virtual Private Networks VPN, configuration of VPN network and related protocols. (2 hrs)
Fault tolerance,exercices about RAID standart levels (3 hrs)
Network monitoring and troubleshooting, Identify symptoms and problems using ICMP, SNMP protocols. Physical issues, logical issues, TCP/IP configuration and command-line utilities. (3 hrs)
Wireless Networking, Characteristics of wireless communication, implementing a wireless network, securing wireless networking. (2 hrs)
Network Security, inside network security, the efect of viruses in networks, outside attacks,physical network security,logical network security , firewalls , proxy servers , Authentication methods, understanding network security threats (4 hrs)
Routing,Elements of a router, basic router configuration,router interfaces configuration,configuration of RIP, IGRProuting protocols. (6 hrs)
Active Directory Services (12 hrs)
Mail service, DNS service, DNS zones, DHCP service.
Definition of domain, Domain Controller, active directory tree and forest structure
Configuring sites ,configuring replication and active directory optimization
Organization units, active directory security policies,group policy objects (GPO) and their configuration.
Backup and recovery of active directory, restoring active direcory

Developed Skills

Students at the end of this course will be able to :
Understand networks and important protocols used
Understand network design,configuration and troubleshooting.
have knowledge of implementing/monitoring network-services and network security

Literature
Writing lectures:
FEJZAJ, Julian. Networks.

Recommended literature:

Mike Harwood
CompTIA® Network+ (N10-004) Cert Guidem,2010

Wendell Odom
CCNA Exam Certification Guide , 1999

Will Panek , James Chellis
MCTS Windows Server 2008 Active Directory Configuration

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Language theory

Teaching Activities	Lectures	Exercises	Lab. Work	Practice	Total
Frequentation Requirement	Non mandatory	75%			0
Teaching Hours	30	30		0	60
Individual Study Hours	65				
Language of Instruction	Albanian				
Evaluation method	Exam				
ECTS Credits	**5**				
Teaching Methods	Year III, Semester I, 15 weeks: 2 hours lectures, 2 hours seminars/week				
Scoring Details	Only writing, 40% of points obligatory, 40 points mark five, every 10 points evaluation increases one mark.				
Assessment	Final exam	100%			
	Total	**100%**			
Main Instructor	**Dr.Denis SAATÇIU**, *tel:++355672197834, e-mail:denis.saatciu@fshn.edu.al*				

Course Description	The course introduces fundamental concepts of theoretical informatics. The course is focused on formale languages, decision trees, Turing machines, effective algorithms anf their improvment, classes P and NP, etc. It examines non solvable problems and focus on explanations of major differences on computational models.
Objectives	The objective of this course is to give knowledge on basic computational models. Following models will be studied: 1. Finite automata and regular expressions 2. Push-down automata and free-context languages 3. Turing machines, introduction to computation, algorithms, Church-Markov-Turing thesis
Preliminary Requirements	Kowledge on discrete mathematics(logic, induction, number theorz, graphics, recursion, verification methods), algorithms and data structures. Knowledge on programming languages.

Lecture Topics

1. **Basic definitions.** Modelling in theoretical computer science. Logic, sets, relations and functions. **(2 hrs).**
2. **Alphabets, strings and languages. (2 hrs).**
3. **Finite automata.** Deterministic finite automaton. NonDeterministic finite automaton. (2 hrs).
4. **Spontaneus transition.** Closure properties. **(2 hrs).**
5. **Regular expressions.** Structural properties of regular languages. **(2 hrs).**
6. **Decidability questions.** Automatic verification. **(2 hrs).**
7. **Context-free grammars. Pumping Lemma (2 hrs).**
8. **Push-down automata.** Closure properties. **(2 hrs).**
9. **Transformation in normal forms. (2 hrs).**
10. **Deterministic context-free languages. (2 hrs).**

11. Turing machines. Principles. Configuration, Construction. Types. **(4 hrs).**
12. Grammars. (2 hrs).
13. Non-computable functions. Undecidable problems. Recursive enumerability. Automatic program verification. **(2 hrs).**
14. Computational complexity Kompleksiteti. The classes P and NP. The satisfiability problem for Boolean expressions **(2 hrs).**

Seminar Topics

1. **Exercises**. Modelling in theoretical computer science. Logic, sets, relations and functions. (2 hrs).
2. **Exercises**. Alphabets, strings and languages. (2 hrs).
3. **Exercises**. Deterministic finite automaton. (2 hrs).
4. Exercises. NonDeterministic finite automaton. (2 hrs).
5. **Exercises**. Spontaneus transition. Closure properties. (2 hrs).
6. **Exercises**. Regular expressions. Structural properties of regular languages.. (2 hrs).
7. **Exercises**. Context-free grammars. Pumping Lemma (2 hrs).
8. **Exercises**. Push-down automata. Closure properties. (2 hrs).
9. **Exercises**. Transformation in normal forms. (2 hrs).
10. **Exercises**. Deterministic context-free languages. (2 hrs).
11. **Exercises**. Turing machines. Principles. Configuration, Construction. Types. (4 hrs).
12. **Exercises**. Grammars. (2 hrs).
13. **Exercises**. Non-computable functions. Undecidable problems. Recursive enumerability. Automatic program verification. (2 hrs).
14. **Exercises**. Computational complexity Kompleksiteti. The classes P and NP. The satisfiability problem for Boolean expressions. (2 hrs).

Developed Skills

After finishing the subject, the student must be capable of:
1. Build finite determinstic and nondeterministic automata for simple problem solving and do transition fromone form to the other and also between automata and regular languages.
2. Define complexity of algorithms
3. Define language regularity
4. Construct minimal automata
5. Understand and build free-context grammars
6. Build push-down automata
7. Use deterministic push-down automata to parse language strings
8. Use Turing Machines
9. Use inductive mathematical techniques to prove regularitz of formal languages and automata theory

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Literature

Writing lectures:

SAATÇIU, Denis. Language Theory.

Recommended leterature:

John Hopcroft; Rajeev Motwani; Jeffrey D. Ullman. Introduction To Automata Theory Languages , and Computation. Second Edition

V. Claus; E.-R. Olderog. Foundations of Theoretical Computer Science. Lecture notes.

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Logical systems

Teaching Activities	Lectures	Exercises	Lab. Work	Total
Frequentation Requirement	Non mandatory	75%	100%	0
Teaching Hours	30	30	0	60
Individual Study Hours	60			
Language of Instruction	Albanian			
ECTS Credits	5			
Teaching Methods	Third Year, First Semester, 15 weeks: 2 lecture hours, 2 exercise hours			
Scoring Details	Written exam at the end of the semester, obligation 50% of points, 45-50 points grade 5, every ten points the evaluation is added with one grade			
Assessment	Semestral project	0%		
	Final exam	100%		
	Total	**100%**		
Main Instructor	**MSc. Petrika Manika**			

Course Description	Logical Systems aims to provide students with the basic concepts of combinatorial and sequential case building. During this course will be taught different coding schemas, Boolean algebra, different algorithms for building logical systems, etc.
Objectives	At the end of the course students will possess the necessary practical-theoretical knowledge of logical systems. They will be able to: Analyze optimal systems using various optimization techniques. Conceptualize optimal systems using various optimization techniques. Synthesize optimal systems using various optimization techniques.
Preliminary Requirements	Good knowledge in a high level programming language, preferably C + +.
Lecture Topics	
Communication. Coded messages. Information effectiveness. Decoding. Building codes. Encoding binary data numbers. Tracking codes and error correcting codes. Logical circles. Definitions and symbols. Boolean algebra. Logical algebra claims. Creating and classification of logical functions. Combinatorial logic systems. Binary collector. Minimization of logic functions. Karnaugh map method. Choosing a minimal subgroup of IP and IP tables. Algebraic solution first implicant tables. Example conception of a coders. Iterative logical systems. Logical Completeness. Symmetric functions. Flip-Flops. Counters conception.	

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Sequential Systems. Sequence detector. Simplification of internal conditions and minimization conditions. Table of conditions set. Systems analysis. Experiments with sequential systems. Synchrone Sequential Systems Asynchrone Sequential Systems.
Developed Skills
Logical Systems building. Using optimization techniques. Sequential Systems building.
Literature **Forced Literature:** Ilia Ninka, Logical Systems **Auxiliary Literature:** 1. Fundamentals of Digital Logic with Verilog design, Stephen Brown and Zvonko Vranesic 2. Digital Design, 2nd Edition, M.M. Mano, Prentice Hall. 3. Contemporary Logic Design, R.H. Katz, Addison-Wesley.

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Databases in oracle

Teaching Activities	Lectures	Exercises	Lab. Work	Total
Frequentation Requirement	Non mandatory	75%	100%	0
Teaching Hours	58	58	0	116
Individual Study Hours	109			
Language of Instruction	Albanian			
ECTS Credits	9			
Teaching Methods	Third Year, First Semester, 15 weeks: 2 lecture hours, 2 exercise hours/ week Third Year, Second Semester, 14 weeks: 2 lecture hours, 2 exercise hours/ week			
Scoring Details	Only writing, 45-50 points mark five, every ten points evaluation increase one mark.			
Assessment	Presence and activation	5%		
	Semester projects	25%		
	Final exam	70%		
	Total	100%		
Main Instructor	Dr. Silvana Greca, *mob:++355692670206, e-mail: silvana.greca@fshn.edu.al*			

Course Description	The course introduces the students to the concepts of databases, database modeling, design and implementation. It covers analysis of the data and their modeling technique using ER/EER, relational theory and normalization, SQL, security and integrity of databases and describes the elements of transactions and concurrency. The course will also instruct the Oracle database. The student will learn the database architecture and to program in pl/sql, to manage users, backup and recovery.
Objectives	This course aims to introduce the students to the fundamental concepts and principles that underline modern-day relational database systems and to prepare them to become able to implement a working database. Advanced Programming in PL/SQL, Backup and recovery.
Preliminary Requirements	The student should have knowledge in algorithmic.
Lecture Topics	
Introduction to databases, database users and their evolution. Types of database management systems. (2 hours). Database System Concepts and the Architecture. (1 hours). Data Modeling Using the Entity-Relationship (ER) Model. (3 hours) Data Modeling Using the EER Model. (1 hours) The Relational Data Model and the Relational Algebra. (4 hours). Relational systems, basic SQL. (4 hours). Advanced SQL and aggregate function in SQL. View and index. (2 hours) Joins: full, left, right, outer. (1 hours) Relational Database Design by ER/EER to Relational Mapping. (2 hours)	

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Database Design Theory and Normalization. (4 hours) Transactions and concurrently techniques. (2 hours) Database Security. (2 hours) Algorithms for query processing and optimization. (2 hours) Oracle database architecture. (2 hours) Creating an Oracle Database. Managing the Oracle Instance (2 hours) Configuring the oracle network environment. Creating and managing database objects and data integrity. Data types. (2 hours) Sequences, synonyms. Manipulating large data sets: unconditional insert, insert all, insert first, pivoting insert. Merge and data dictionary. (2 hours) Introduction to PL/SQl. Control structures. (2 hours) Exceptions. Composite data types. (2 hours) Explicit cursors. (2 hours) Procedures and functions. (2 hours) Packages. (2 hours) Triggers. (2 hours) Managing transaction and administering user security. (2 hours) Sql loader. Oracle data pump, expdp and impdp. (2 hours) Backup concepts. Performing database backups. (2 hours) Recovery concepts. Performing database recovery. (2 hours)
Exercises Topics
Discussion about databases, database users and their evolution. Types of database management systems and their advantages. (2 hours) Discussion about DBMS, data models, schemas. Architecture with three levels. Database languages. (2 hours) ER exercises. (2 hours) ER complex exercises. EER exercises. (2 hours) Relational algebra exercises. (2 hours) Relational algebra complex exercises. (2 hours) Exercises with DDL and constraints in SQL Plus (2 hours) Sql exercises. (2 hours) Sql complex exercises and with joins. (2 hours) Exercises with set operators, subqueries. (2 hours) Exercises in normalization. (2 hours) Different exercises in normalization. (2 hours) Exercises with transactions. (2 hours) Security management. (2 hours) Optimization exercises. (2 hours) Oracle, it's architecture and storage. (2 hours) Exercises creating database. (2 hours) Exercises, managing database objects and constraints. (2 hours) Exercises for manipulating large data sets. (2 hours) Exercises for PL/SQl control structures. Records and collections in PL/SQL. (2 hours) Exercises for exception and explicit cursors. (2 hours) Exercises for procedures and functions. (2 hours) Exercises for packages. (2 hours) Exercises for triggers instead of, before and after. (2 hours) Exercises for triggers, transactions, locks. (2 hours) Exercises for user management, role and privileges. (2 hours) Exercises for import/export and sql loader. (2 hours)

Complex exercises. (2 hours) Exercises for backup and recovery of Oracle DB. (2 hours)
Developed Skills
At the end of this course, students will be able to describe the databases as a special type of software for storing, receiving and manipulating large amounts of data, relational models and implementation techniques. They gain practical skills for data analysis and design of relational databases using ER or EER diagrams. Students will be able to apply normalization techniques, to use SQL for creating and manipulating database and PL/SQL as a language to manage data in Oracle. They will be able to handle the security, backup and recovery processes.
Literature Required Literature: "Sistemet e bazave të të dhënave", 2015,Silvana Greca, Alba Çomo. Lecture or presentation from the lecturer. Additional literature: Date C.J. "An Introduction to Database Systems 8Edition" Addison Wesley. Oracle Database 11g PL/SQL Programming, Michael McLaughlin OCA Oracle Database 11g Administration I Exam Guide (Exam 1Z0-052) Beginning Oracle Database 11g Administration: From Novice to Professional

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

GIS

Teaching Activities	Lectures	Exercises	Lab. Work	Total
Frequentation Requirement	Non mandatory	75%	100%	0
Teaching Hours	30	30	0	60
Individual Study Hours	44			
Language of Instruction	Albanian			
ECTS Credits	4			
Teaching Methods	Third Year, Second Semester, 14 weeks: 2 lecture hours, 2 exercise hours			
Scoring Details	Continous Check, 50% of overall points are mandatory, 45-50 points five grade, every ten points the estimation is added by one grade.			
Assessment	Participating	5%		
	Middle term	45%		
	Final exam	50%		
	Total	**100%**		
Main Instructor	**Prof. Asoc. Dr. Endri Xhina,** *e-mail:endrit.xhina@fshn.edu.al*			

Course Description	This course is focused on introducing students with the usage of geographic information systems. Students learn GIS evolution, architecture, layers, etc. Technical topics that are covered include integration of data from a variety of sources, database structure and design issues, spatial analysis capabilities, data quality and data documentation.
Objectives	• Understand the data creation process and create simple data sets and/or add to existing data. • Perform basic spatial analyses. • Create high-quality maps and associated graphics and text that clearly communicate spatial information and analyses.
Preliminary Requirements	Excellent knowledge in: Algorithmic, Data Structures, Java, C++, Web Programming.

Lecture Topics

1. **Introduction to Digital Geography.**Learning Objectives.GIS Definition. A brief history to GIS. GIS as a Growth Industry. Sample Application Areas of GIS. The Study of GIS.**(2h)**
2. **Digital geographic data and maps.** Developing Spatia Awareness. Spatial Measurement level. Spatial Location and reference. Spatial patterns. Geographical data collection.Populations and sampling schemes. **(2h).**
3. **Map Basics.** Abstract nature of maps. Map scale. Map projections. Grid systems for mapping. Map symbolism. Map abstraction and catrography databases. **(2h).**
4. **Gis computer structure basics.** Basic computer file structures. Database management structures. **(2h).**
5. **Gis data model.** Graphic representation of entitites and attributes. Gis system data models. Raster models. Raster surface models. Compact storing of raster data. Vector model. An object oriented data model. A vector model to represent surfaces. System models. **(2h).**
6. **GIS Input.** Input devices. Methods of vector input. Methods of raster input. Remote sensing data input. Gps data input. Secondary data. **(2h).**
7. **Data storage and editing.** Gis database storage. Basic error types. Error detection and editing.

FACULTY OF NATURAL SCIENCES
DEPARTAMENT OF INFORMATICS

Entity error: Vector. Attribute Errors: Raster and Vector. Joining adjacent maps: Edge matching.**(2h).**

8. **Query and description.** Model flowcharting. Gis data query. Locating and identifying spatial objects. Defining spatial characteristics.Point attributes. Line attributes. Area attributes. Working with higher level objects.**(2h).**
9. **Measurements.** Measuring Length. Measuring polygons. Measuring shape. Measuring distance**(2h).**
10. **Classification.** Classification principles. Elements of reclassification. Neighborhood functions. Buffers.**(2h).**
11. **Statisticals Surfaces.** What are surfaces? Surface mapping. Nontopographical surfaces. The DEM. Interpolation.**(2h).**
12. **Terrain Analysis**. Terrain reclassification. Cut and fill.**(2h).**
13. **Spatial Arrangement**. Point,area and line arrangements. Point patterns. Nearest neighbor analiysis. Area patterns. Direction and circular statistics. Gravity model. Routing and allocation.**(2h).**
14. **Cartography and Visualisation**. The display of analysis. Cartographic output. **(1h).**
15. **GIS design**. Application design. Analytical model design. Database design. System design. System implementation. **(1h).**

Developed Skills

1. Students are introduced with geographic information systems in theory and practice.

Literature

Written lectures:

Xhina, Endri. GIS.

Recommended Literature:

1. "Fundamentals of Geographic Information Systems Fourth Edition" **Michael N. Demers**
2. http://ocw.mit.edu

FACULTY OF NATURAL SCIENCES
DEPARTMENT OF INFORMATICS

WEB PROGRAMMING

Teaching Activities	Lectures	Exercises	Lab. Work	Practice	Total
Frequentation Requirement	Non mandatory	75%	100%	0%	0
Teaching Hours	58	32	15	0%	105
Individual Study Hours	153				
Language of Instruction	Albanian				
ECTS Credits	**10**				
Teaching Methods	Third Year, First Semester, 15 weeks: 2 lecture hours, 1 hour exercise, 1 hour lab. work. Third Year, Second Semester, 14 weeks: 2 lecture hours, 2 exercise hours.				
Scoring Details	Laboratory work 15%, group project 35%, final exam 50%. 50% of overall points are mandatory in final exam. Final evaluation, 45-50 points five grade, every ten points the estimation is added by one grade.				
Assessment	Laboratory work	15%			
	Web Project	35%			
	Final exam	50%			
	Total	**100%**			
Main Instructor	**MSc. Ana Dhembi,** *e-mail:ana.dhembi@fshn.edu.al*				

Course Description	Web programming course is composed by a set of lectures that aim to introduce students with the development and implementation of web applications based on open-source technologies.
Objectives	Students are presented with web technologies, the architecture of web systems, and how to develop functional and usable web applications. By participating in the course project, they work in small groups to develop their own web application. This task aim to help students share and coordinate their work within a small group, and put into practice the knowledge given about steps that should follow to successfully implement the app. Some of the main technologies they study and use in their projects are, in client-side programming HTML, JS, Jquery ect, and server-side PHP, MySQL. Additional tools and frameworks build on these technologies are presented during the second semester of this course.
Preliminary Requirements	Knowledge in high level programming languages and database systems.

Lecture Topics
(Exercises and lab. work are based on lecture topics)

1. **Client Side Programming (p. L1. 2-38) (4 h).** Introduction to HTML. Tables, forms, form elements, lists, hyperlinks, images etc. CSS properties.
2. **Javascript (p. L1. 39-56) (2h).** Introduction to Javascript. Statements in Javascript. Events. Functions. Objects. JS applications.
3. **DOM and Jquery (p. L1. 58-68) (2h).** Introduction to DOM. Manipulate DOM elements using methods and properties. Jquery syntax. Programming elements used in JQuery.
4. **Server Side programming (p. L1. 70-74) (2h).** Client Server Architecture. Introduction to PHP. Variables, Operators, Data types.
5. **Statements in PHP (p. L1. 75-82) (2h).** If-Else statement. FOR,WHILE,DO WHILE

statements. SWITCH-CASE statement.

6. **Strings in PHP (p. L1. 83-88) (2h).** Declaration and strings manipulation in PHP. String functions.
7. **Functions in PHP(p. L1. 89-96) (2h).** Functions and code module in PHP. Recursive functions. Including files inside other files in PHP.
8. **Browser communication (p. L1. 97-103)(2h).** Introduction to communication client server protocols. HTTP Request, HTTP Response structure. Sending data through $_REQUEST, $_POST, $_GET arrays. Differences.
9. **Files in PHP . Array's in PHP (p. L1. 103-116) (2h).** Open, read, write files in PHP. Declaration and usage of arrays in PHP.
10. **Cookies and Sessions (p. L1. 117-124)(2h).** Create and manage cookies and sessions. Delete cookies and sessions.
11. **PHP connection to MYSQL (p. L1. 125-155) (4h).** Introduction to databases. Data types, operators. Functions. Data manipulation.
12. **AJAX Technology (L2.1) (4h).** Ajax architecture. Introduction to JQuery library. Ajax applications.
13. **Multi-Layer Architecture (L2.2) (4h).** Client-Server Architecture. Three layer architecture and MVC model, MVV model. N-layers architecture.
14. **PHP tools and frameworks (L2.3) (6h).** Introduction to tools and other web technologies used in multi-layer architecture. Using PHP and JS frameworks to build web systems.
15. **System Diagrams. (4h).**
16. **Database relational schema. (4h).**
17. **Layout design. (2h).**
18. **Project presentations. (8h).**

Developed Skills

1. Web programming skills – client side programming, server-side programming.
2. Programming knowledge in PHP, Javascript, AJAX, Jquery, MySQL.
3. Building web app using PHP and JS frameworks.
4. Skills in developing and managing real-life projects.

Literature

Base literature and books:

L1 - Xhina, E. Programim në Web, SHBLU 2010.

L2 – Dhembi, A. Leksione të shkruara.

Recommended literature:

1. http://web.mit.edu/
2. Software Engineering for Internet Applications by Eve Andersson, Philip Greenspun, and Andrew Grumet MIT Press 2006; ISBN 0262511916
3. SQL for Web Nerds by Philip Greenspun
4. **Pressman S. R.** Software Engineering A practitionier's Approach. Seventh Edition.
5. **Bruegge B., Dutoit A.** Object Oriented Software Engineering. Conquering Complex and Changing Systems.

FACULTY OF NATURAL SCIENCES
DEPARTMENT OF INFORMATICS

SOFTWARE ENGINEERING

Teaching Activities	Lectures	Exercises	Lab. Work	Total
Participation	Non mandatory	75%	100%	0
Teaching Hours	28	14	0	42
Individual Study Hours				
Teaching Language	Albanian			
ECTS Credits	5			
Teaching Methods	Third Year, Second Semester, 14 weeks: 2 lecture hours, 2 hours of tutorial			
Grading System	Final Exam, 50% of overall points are mandatory, 45-50 points five grade, every ten points the estimation is added by one grade.			
Assessment	Software engineering project	20%		
	Final exam	80%		
	Total	**100%**		
Main Instructor	**Bora BIMBARI**			

Course Description	This course consist of a set of lectures that provide students with insight to software developing using methodologies and technologies of web programming. It provides a sound foundation on software engineering as well.
Objectives	The course aims to make students gain knowledge on the characteristics of software development projects, software lifecycle, and the activities to be followed to result in successful software. During the course the students must develop a software as part of team, so that they can apply the principles covered during the course through every phase of the software lifecycle. They are required to meet deadlines and to use several web programming technologies.
Preliminary Requirements	Programming fundamentals (and at least one programming language). Database systems.

Lecture Topics

1. **Introduction to software engineering (2h).** Software crisis. Characteristics of software development projects and common reasons for their failure. Software development as an engineering activity.

2. **Software development process (2h).** Concept of software development process, software lifecycle. The necessity of following a process, activities of the software development process.

3. **Software development process models (4h).** Process modelling. Choosing a process model that is suitable for the project. Linear process model, Waterfall process model, RAD, prototyping model, evolutionary models.

4. **Agile software development (2h).** Agile development manifesto. Extreme programming,

Microsoft's process model.

5. **Software development project (2h).** Introduction to project management, properties of software development projects.

6. **Requirements elicitation and management (4h).** Software requirement definition, functional and nonfunctional requirements. Requirements elicitation activities. Requirements analysis and modeling.

7. **Software design (6h).** Introduction to software modelling. Deriving software model from the requirements model. Software architecture. Software architecture vs software modelling. Design patterns. Data modelling. Graphical user interface modelling. Object oriented modelling.

8. **Coding (2h).** Coding principles and standards. Refactoring.

9. **Software testing.** Software testing principles. Testing and software quality assurance. Types of testing (system testing, stress testing, load testing, etc.). Testing techniques (black box, white box).

10. **Software maintenance.** Introduction to maintenance principles. Maintenance types.

11. **Software quality assurance.** Software quality assurance activities.

Knowledge and skills developed during the course

1. Software lifecycle. Engineering processes and application of engineering principles to software development.
2. Software project management.
3. Web programming principles – client side programming, server side programming.
4. Programming in PHP, Javascript, AJAX, JQuery, MySql.

Text book

Pressman S. R. Software Engineering A practitioner's Approach. Seventh Edition.
Bruegge B., Dutoit A. Object Oriented Software Engineering. Conquering Complex and Changing Systems.

Lecture notes, **BIMBARI Bora**