Emri i Lëndës : Hyrje në Algoritme dhe Programim							
Kodi	**Tipi**	**Semestri**	**Leksione (orë/javë)**	**Seminare (orë/javë)**	**Lab (orë/javë)**	**Kredite**	**ECTS**
EMS 121	D	Vjeshtë	3.00	1.00	0.00	3.50	5.00

Lektori	Jora Banda, Msc
Asistenti	Edlir Spaho, MSc
Gjuha e kursit	Anglisht
Niveli i lëndës	Bachelor
Përshkrimi	This course provides an introduction to mathematical modeling of computational problems. It covers the common algorithms, algorithmic paradigms, and data structures used to solve these problems. The course emphasizes the relationship between algorithms and programming, and introduces basic performance measures and analysis techniques for these problems.
Objektivat	At the end of the course, the student is able to analyze a problem, formulate an algorithm for solving it, and code such an algorithm into an executable program. The student understands some basic algorithms and a general-purpose programming language.
Konceptet Kryesore	- Computer System - Input/Output - Programming (Programming Language, programmer, program) - Algorithm, flowchart, pseudocode - Decision-making process - Loops - Functions - Arrays

Programi i Lëndës

Java	Tema
1	Introduction to Computers and Programming (pg. 1-22) This chapter provides an introduction to the field of computer science and covers the fundamentals of programming, problem solving, and software design. The components of programs, such as key words, variables, operators, and punctuation are covered. The tools of the trade, such as pseudocode, flow charts, and hierarchy charts are also presented.
2	Introduction to Algorithms, Flowcharts and Pseudocode (pg. 27-73) In this topic, it will be discussed over the algorithm. Students will be presented with flowcharts and pseudocodes. At the same time, it will be explained the rules needed to design a flowchart and pseudocode.
3	Introduction to C++ (pg. 27-73) This chapter gets the student started in C++ by introducing data types, identifiers, variable declarations, constants, comments, program output, simple arithmetic operations, and C-strings. Programming style conventions are introduced and good programming style is modeled here, as it is throughout the text.
4	Expressions and Interactivity (pg. 83-132) his chapter the student learns to write programs that input and handle numeric, character, and string data. The use of arithmetic operators and the creation of mathematical expressions are covered in greater detail, with emphasis on operator precedence. Debugging is introduced, with a section on hand tracing a program.
5	Making Decision (pg. 149-211) In this chapter, student learns about relational operators, relational expressions and how to control the flow of a program with the if, if / else, and if / else if statements. The conditional operator and the switch statement are also covered. Crucial applications of these constructs are covered, such as menu-driven programs and the validation of input.
6	Making Decision (pg. 149-211) In this chapter, student learns about relational operators, relational expressions and how to control the flow of a program with the if, if / else, and if / else if statements. The conditional operator and the switch statement are also covered. Crucial applications of these constructs are covered, such as menu-driven programs and the validation of input.

7	Making Decision (pg. 149-211) In this chapter, student learns about relational operators, relational expressions and how to control the flow of a program with the if, if / else, and if / else if statements. The conditional operator and the switch statement are also covered. Crucial applications of these constructs are covered, such as menu-driven programs and the validation of input.
8	Midterm
9	Loops and Files (pg. 227-284) This chapter covers repetition control structures. The while loop, do - while loop, and for loop are taught, along with common uses for these devices. Counters, accumulators, running totals, sentinels, and other application-related topics are discussed. Sequential fi le I/O is also introduced. The student learns to read and write text fi les, and use loops to process the data in a file.
10	Loops and Files (pg. 227-284) This chapter covers repetition control structures. The while loop, do - while loop, and for loop are taught, along with common uses for these devices. Counters, accumulators, running totals, sentinels, and other application-related topics are discussed. Sequential fi le I/O is also introduced. The student learns to read and write text fi les, and use loops to process the data in a file.
11	Functions (pg. 299-361) In this chapter the student learns how and why to modularize programs, using both void and value returning functions. Argument passing is covered, with emphasis on when arguments should be passed by value versus when they need to be passed by reference. Scope of variables is covered, and sections are provided on local versus global variables and on static local variables. Overloaded functions are also introduced and demonstrated.
12	Functions (pg. 299-361) In this chapter the student learns how and why to modularize programs, using both void and value returning functions. Argument passing is covered, with emphasis on when arguments should be passed by value versus when they need to be passed by reference. Scope of variables is covered, and sections are provided on local versus global variables and on static local variables. Overloaded functions are also introduced and demonstrated.
13	Arrays (pg. 377-429) In this chapter the student learns to create and work with single and multidimensional arrays. Many examples of array processing are provided including examples illustrating how to find the sum, average, highest, and lowest values in an array and how to sum the rows, columns, and all elements of a two-dimensional array. Programming techniques using parallel arrays are also demonstrated, and the student is shown how to use a data fi le as an input source to populate an array.
14	Arrays (pg. 377-429) In this chapter the student learns to create and work with single and multidimensional arrays. Many examples of array processing are provided including examples illustrating how to find the sum, average, highest, and lowest values in an array and how to sum the rows, columns, and all elements of a two-dimensional array. Programming techniques using parallel arrays are also demonstrated, and the student is shown how to use a data fi le as an input source to populate an array.
15	Introduction to Classes (pg. 711-785) The student now shifts focus to the object-oriented paradigm. This chapter covers the fundamental concepts of classes. Member variables and functions are discussed. The student learns about private and public access specifications, and reasons to use each. The topics of constructors, overloaded constructors, and destructors are also presented.
16	Final Exam

Parakushtet	Studenti duhet të frekuentojë lëndën në masën minimale prej 75%.
Literatura	• Tony Gaddis, "Starting out with C++ : From control structures through objects", 8th Edition, Pearson, 2015
Referenca të tjera	• Deitel H. and Deitel P., “C++ How to Program”, 10th Edition, Prentice Hall, 2017

Mënyra e Vlerësimit të Lëndës

Notat e Ndërmjetme	Sasia	Përqindja
Gjysmë finale	1	30
Kuize	0	0
Projekte	0	0
Projekte semestrale	0	0
Punë laboratori	1	10
Pjesëmarrja në mësim	0	0
Kontributi i notave të ndërmjetme mbi vlerësimin final		**40**
Kontributi i provimit final mbi vlerësimin final		**60**
Total		**100**

Ngarkesa ECTS (Në Bazë të Ngarkesës së Studentit)

Aktivitetet	Sasia	Kohëzgjatja (orë)	Ngarkesa Totale (orë)
Kohëzgjatja e kursit (Duke përfshirë edhe javën e provimeve : 16x Orët totale të kursit)	16	4	64
Orët e studimit jashtë klase (Parapërgatitje, Praktika etj)	14	4	56
Detyra	1	0	0
Gjysmë finale	1	4	4
Provimi final	1	8	8
Të tjera	0	0	0
Ngarkesa totale e orëve			**132**
Ngarkesa totale e orëve / 25 (orë)			**5.28**
ECTS			**5.00**