



REPUBLIKA E SHQIPËRISË

“ISMAIL QEMALI” UNIVERSITY
FACULTY OF TECHNICAL AND NATURAL SCIENCES
COMPUTER SCIENCE DEPARTMENT

SYLLABUS
OBJECT-ORIENTED PROGRAMMING IN C++

Curriculum Developer

PhD. Eljona Proko

Ph.D. in Informatics

Credits and hours	Credits	Hour in the classroom						Individual study	Total
		Lectures	Seminars	Labs	Practice	Course project	Total		
	6 ECTS	45	15	15	0	0	75	75	150
Course Typology		B - Characteristic disciplines							
Academic year /semester		2026-2027/ Second year, first semester							
Nature and requirement of the course		Mandatory course							
Study program		Bachelor in Computer Engineering							
Course Code		CS231E							
Email of curriculum developer/lecturer		eljona.proko@univlora.edu.al							

SUMMARY AND LEARNING OUTCOMES

Course Description	This course introduces students to C++ programming language, a dominant language in today's industry. C++ language is a superset of the “C” language and was initially known as “C with Classes”. C++ is a general-purpose programming language used to create software with a wide variety of applications. It runs on many platforms like Windows, Linux, Unix, Mac, etc. In this course, we will learn the mechanics of editing and compiling programs in C++. We will begin by discussing the essential elements of C++ programming: variables and operators, conditions and loops, functions, arrays, pointers, structures, and string classes. Then, we will cover the basics of object-oriented programming: classes, inheritance, templates, exceptions, and file manipulation.
Course Objectives	▪ To understand the basic syntax and structure of a C++ program. ▪ To write and compile basic C++ programs. ▪ To create and use simple functions, arrays, pointers, and structures. ▪ To gain a basic understanding of the concepts of classes and objects. ▪ To understand and apply the principles of encapsulation, inheritance, and polymorphism. ▪ To design and implement class hierarchies. ▪ To use virtual functions and abstract classes. ▪ To understand and use constructors, destructors, and copy constructors.



REPUBLIKA E SHQIPËRISË

“ISMAIL QEMALI” UNIVERSITY
FACULTY OF TECHNICAL AND NATURAL SCIENCES
COMPUTER SCIENCE DEPARTMENT

Expected Outcomes	At the end of the course, students will be able to: □ compile and execute code written in C++; ▪ code using elementary data types and conditional and iteration structures; ▪ define and use functions, arrays, structs, unions, and enumerations; ▪ write C++ applications using principles of object-oriented programming; ▪ write templates and manipulate files; ▪ translate simple word problems into C++; ▪ perform debugging and fixing of common C++ errors; and ▪ manage memory appropriately, including proper allocation/reallocation procedures.
FUNDAMENTAL CONCEPTS ▪ Conditions, loops ▪ Functions, arrays ▪ Structures, pointers ▪ Classes, objects ▪ Encapsulation, inheritance, polymorphism, and abstraction ▪ Templates ▪ Exceptions ▪ File manipulation	

Week	COURSE TOPICS	Reference Literature	Teaching Methodology
I.	Topic 1 – Introduction to the CS231E Syllabus. Introduction to C++ Language. Topic 2 – A Simple C++ program. Organizing C++. Variables, simple I/O. Comments. Types. Expressions. Topic 3 – Tokens. Language operators. Inter-operator priority. Seminar - Exercises on C++ programs, examples with operators: arithmetic, logical, relational, etc. Laboratory - Laboratory exercises on operators.	Syllabus CS231E [1] page. 135-168 [1] page. 273-303	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminar 1 hour Laboratory 1 hour
II.	Topic 4 – Conditions: if, switch. Topic 5 – for loop, continue, break, go to, and return. Topic 6 – while and do-while loops. Seminar - Exercises on instructions: if, switch, while, do, for, etc. Laboratory - Laboratory exercises on conditions and loops.	[1] page. 225-239	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminar 1 hour Laboratory 1 hour
III.	Topic 7 – Function prototype. Parameters and arguments. Local and global variables. Topic 8 – Function call by value, by reference, by pointer. Numerations. Inline functions. Topic 9 – Recursion. Math library functions. Seminar - Exercises on simple functions, Math library functions, inline, recursive functions, factorial examples, raising a number to powers, etc. Laboratory - Laboratory exercises on functions.	[1] page. 305-340	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminar 1 hour Laboratory 1 hour



REPUBLIKA E SHQIPËRISË

“ISMAIL QEMALI” UNIVERSITY
FACULTY OF TECHNICAL AND NATURAL SCIENCES
COMPUTER SCIENCE DEPARTMENT

Week	COURSE TOPICS	Reference Literature	Teaching Methodology
IV.	Topic 10 – Arrays, two-dimensional arrays. Topic 11 – Pass array to functions. Topic 12 – Pass 2D array to functions. Seminar - Exercises on arrays, searching for an element, sum, max, min. Sorting elements in an array. Laboratory - Laboratory exercises on arrays.	[1] page. 171-181	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminar 1 hour Laboratory 1 hour
V.	Topic 13 – Strings in C++. String functions. Topic 14 – Arithmetic of pointers. Pointers and arrays. Topic 15 – Array of pointers. Function pointers. Dynamic memory. Seminar - Exercises on strings, pointers, functions, and arrays. Laboratory - Laboratory exercises on strings and pointers.	[1] page. 182-198	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminar 1 hour Laboratory 1 hour
VI.	Topic 16 – Structures. Declaring and initializing variables of structs. An array of structures. Topic 17 – Function in structs (member functions). Passing struct to function (non-member functions). Topic 18 – Pointers to structures. Nested structures. Seminar - Exercises on structures, functions, arrays, and pointers. Laboratory - Laboratory exercises on structures.	[1] page. 201-224	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminar 1 hour Laboratory 1 hour
VII.	Topic 19 – Classes. The general form of classes. Defining member functions inside or outside the class definition. Topic 20 – Other forms of variable initialization. Using private variable values. Topic 21 – Calculations with class variables. Using functions within a class. Seminar - Exercises on classes, class Point, class Circle, Triangle, etc. Laboratory - Laboratory exercises in classes.	[1] page. 449-478	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminar 1 hour Laboratory 1 hour
VIII.	Topic 22 – Default, parameterized, and overloaded constructors. Topic 23 – Empty and copy constructors. Destructors. Topic 24 – Pointer to C++ classes. Seminar - Exercises on classes, constructors, destructors and pointers. Laboratory - Laboratory exercises on constructors and destructors.	[1] page. 571-582	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminar 1 hour Laboratory 1 hour
IX.	Topic 25 – Friend functions and classes. This pointer. Topic 26 – Pointers to overloaded functions. Overloaded constructors. Topic 27 – Operator overloading, +, -, <<, >>, [], etc. Seminar - Exercises on friend functions and classes, pointers, and operator overloading.	[1] page. 527-547	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminar 1 hour



REPUBLIKA E SHQIPËRISË

“ISMAIL QEMALI” UNIVERSITY
FACULTY OF TECHNICAL AND NATURAL SCIENCES
COMPUTER SCIENCE DEPARTMENT

Week	COURSE TOPICS	Reference Literature	Teaching Methodology
	Laboratory - Laboratory exercises on friend functions and classes, and operator overloading.		Laboratory 1 hour
X.	Topic 28 – Class inheritance. Different forms of inheritance. Accessibility of base class members in the derived class. Single inheritance. Topic 29 – Multi-level inheritance, hierarchical inheritance, and multiple inheritance. Topic 30 – Constructors and destructors in derived classes. Pointers in derived classes. Seminar - Exercises on inheritance, constructors, destructors, and pointers in derived classes. Laboratory - Laboratory exercises on inheritance.	[1] page. 583-610	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminar 1 hour Laboratory 1 hour
XI.	Topic 31 – Virtual functions, pure virtual functions, abstract base classes. Topic 32 – Run-time polymorphism. Function overriding. Topic 33 – Virtual base classes. Seminar - Exercises on abstract and virtual base classes. Laboratory - Laboratory exercises on polymorphism.	[1] page. 613-639	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminar 1 hour Laboratory 1 hour
XII.	Topic 34 – Template functions. Overloading template functions. Topic 35 – Template classes. Topic 36 – Template functions and classes with multiple parameters Seminar - Exercises on template functions and classes. Laboratory - Laboratory exercises on template functions and classes.	[1] page. 665-698	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminar 1 hour Laboratory 1 hour
XIII.	Topic 37 – Sequential access files. Read and write operations. Accessing files in a loop. Topic 38 – Texts in files. Texts and numbers in files. Writing and reading characters. Topic 39 – Reading lines. Ways to open files. Seminar - Exercises on text files. Laboratory - Laboratory exercises on text files.	[1] page. 1073-1080	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminar 1 hour Laboratory 1 hour
XIV.	Topic 40 – Position in the file. Direct access files (binary files). Texts and numbers in files. Topic 41 – File Handling through C++ classes. Topic 42 – Objects in files. Seminar - Exercises on text and binary files. Laboratory - Laboratory exercises on binary files.	[1] page. 1081-1107	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminar 1 hour Laboratory 1 hour



REPUBLIKA E SHQIPËRISË

“ISMAIL QEMALI” UNIVERSITY
FACULTY OF TECHNICAL AND NATURAL SCIENCES
COMPUTER SCIENCE DEPARTMENT

Week	COURSE TOPICS	Reference Literature	Teaching Methodology
XV.	<i>Topic 43 – Projects in C++.</i> <i>Topic 44 – Organizing classes in projects.</i> <i>Topic 45 – Organizing classes in projects.</i> <i>Seminar – Organizing C++ projects.</i> <i>Laboratory - Laboratory exercises on C++ projects.</i>		Lecture 1 hour Lecture 1 hour Lecture 1 hour <i>Seminar 1 hour</i> <i>Laboratory 1 hour</i>

KNOWLEDGE ASSESSMENT							
ATTENDANCE		Attendance for lectures and seminars is mandatory at 75%. Otherwise, the student loses the right to take the exam. Absences are the personal responsibility of the student, as are any missed assignments or tests. A student who has less than 75% attendance for the period relevant to any partial exam, during which they will be tested, will not be allowed to take the corresponding exam and will be graded with an M (Missing). If the student has attended the course but does not show up for the next exam, they will be graded NP (Not Present).					
CONTINUOUS ASSESSMENT	Assessment Method	Assessment Time	Learning Topics to be Assessed	Maximum Points	Knowledge Testing Method	Test Content	
	Partial Exam	Week 7-8	1-15	25 points	Written exam	Concepts and exercises	
	Final Exam	End of the semester	16-45	60 points	Written exam	Concepts and exercises	
	Continuous Assessment*(Course Assignment*)	During the semester	-	15 points	A written report and oral presentation	Assignments and a project	
	* Continuous Assessment – Active participation in seminars (assignments), and a project (students will work individually on creating applications in C++).						
	The conversion of points to the final grade is as follows:						
	0 - 40 points	41 - 50 points	51 - 60 points	61 - 70 points	71 - 80 points	81 - 90 points	91 - 100 points
	4	5	6	7	8	9	10
LITERATURE							
Primary literature		[1] “The C++ programming language”, 4 th edition, Bjarne Stroustrup, Addison Wesley, 2013. ISBN 978-0321563842.					



REPUBLIKA E SHQIPËRISË

“ISMAIL QEMALI” UNIVERSITY
FACULTY OF TECHNICAL AND NATURAL SCIENCES
COMPUTER SCIENCE DEPARTMENT

Recommended literature	○ “Introduction to C++”, 1st edition, George S.Tselikis, CRC Press, 2023. ISBN: 9781003230076. https://doi.org/10.1201/9781003230076 ○ “Fundamentals of C++ Programming”, Richard L. Halterman, School of Computing, Southern Adventist University, 2018. ○ “Modern C++ for Absolute Beginners: A Friendly Introduction to C++ Programming Language and C++11 to C++20 Standards”, Slobodan Dmitrović, 2020.
FINAL REMARKS FROM THE COURSE INSTRUCTOR	
Provisions	Students are expected to develop values and ethics that reflect their interests and behavior from a professional perspective. Students must respect individual differences, be cooperative, and in accordance with ethical rules, they should respect ethnic differences and assist individuals with disabilities with inclusive and supportive attitudes.
Delays	Delays in lectures and seminars will not be tolerated.
Academic integrity and copyright	Each test or written work must be completely individual and original. Plagiarism (mixed combinations) or theft of information through various means will not be allowed.

HEAD OF DEPARTMENT OF COMPUTER SCIENCES
Dr. Alketa HYSO