COURSE DESCRIPTION

Course number	0418-143	**Course Title**	Fundamentals of Computer Programming
Credit hours	4 (3:3:4)	**Course Coordinator**	Dr. Mohamed Smaoui
Compulsory	No	**Prerequisite**	0418-101

Catalog Description

Overview of programming concepts. Repetition techniques. Functions, recursion, parameter passing mechanisms. Data structures, programmer-defined structures, programmer-defined types, classes, and objects. I/O concepts, including file processing. Basic problem-solving techniques and design of simple algorithms.

Textbook

Title*:* *C++ How To Program*
Author: *Paul Deitel and Harvey Deitel*
Publisher: *Pearson, Latest Edition*

Major Topics Covered in the Course

	Topic	**Hours**
1	Introduction to C++, variable declaration, basic data types, C++ syntax, conditional statements, loops	3
2	Converting Python to C++, problem solving in both python and C++	3
3	Arrays, vectors, lists, multidimensional arrays, sets, nested loops	6
4	Library functions, programmer-defined functions, parameter passing mechanisms, function overloading, type inference (auto keyword)	6
5	Recursive functions, simple recursive algorithms, comparison of recursive and iterative methods	6
6	I/O and file processing, text-formatting, and encoding modes	4
7	Introduction to object-oriented programming: classes and objects, encapsulation, composition, inheritance, polymorphism, abstraction, exception handling	8
8	Problem-solving techniques such as searching and sorting algorithms	6
	Total	**42**

Laboratory work

Week	Topic
1	Programming intro, problem-solving, Variables and types
2	Conditionals and loops + 1 Mapping problem from Python to C++
3	Loops & arrays (Introduction) + 1 Mapping problem from Python to C++
4	Arrays (Advanced Topics)
5	Functions (Introduction) + 1 Mapping problem from Python to C++
6	Functions (Advanced Topics)
7	Recursion (Basics)
8	Recursion (Applications)
9	Input and output (I/O) (Basics)
10	Input and output (I/O) (Advanced Topics)
11	Classes and objects (Introduction)
12	Classes and objects (Encapsulation)
13	Classes and objects (Composition, Inheritance, Polymorphism)
14	Classes and objects (Review and Applications)

Grading

20%	Midterm 1
20%	Midterm 2
15-20%	Lab
0-5%	Hmk/quiz
40%	Final

Course Outcomes

Students should be able to:

1. Analyze and explain the behavior of simple and intermediate programs.
2. Understand and implement methods to structure programs into simple, reusable code using object-oriented programming concepts.
3. Design, implement, test, and debug programs.
4. Understand basic problem-solving techniques.

Relationship between Course Outcomes and Student Outcomes

Course Outcomes	CS SOs					
	1	2	3	4	5	6
1.	H					
2.		H				
3.		H				
4.	H					

Last updated in May 2025.