

Course Title	Introduction to Programming and Object Orientation				
Course Code	CSC600				
Course Type	Compulsory				
Level	Master (2 nd Cycle)				
Year / Semester	1 st year / 1 st semester				
Teacher's Name	TBA				
ECTS	10	Lectures / week	3 Hours / 14 weeks	Laboratories / week	N/A
Course Purpose and Objectives	<p>The course aims to Introduce students to basic concepts of the computer science discipline from theoretical concepts and areas of study to the role of computer scientists in today's society.</p> <p>A primary objective of the course is to introduce fundamental ideas of problem solving and structured programming, using the principles of top-down design, stepwise refinement and procedural abstraction.</p> <p>Additionally, the course will introduce object oriented programming as a modelling tool. Object orientation principles and common practices will be delivered thus enabling the student to be able to create their own data types.</p>				
Learning Outcomes	<p>Upon successful completion of this course student will be able to:</p> <ul style="list-style-type: none"> • Describe and discuss the key theoretical concepts of the Computer Science discipline. List the key areas of study in Computer Science and discuss their interrelationships. • State the importance of the Computer Science discipline. Argue about the role and ethical responsibility of Computer Scientists in our society. • Define and choose and use suitable primitive data types in basic statements and write programs containing selection and repetition control constructs. • Specify and implement functions and use them as an abstraction mechanism to modularize a programmatic solution. • Declare, initialize and manipulate arrays and pointers. Build relatively simple programs utilizing arrays and pointers. • Create and manipulate classes and instance objects. Define data attributes, get and set functions and constructors. • Design and build hierarchy of classes in order to model and solve problems. 				
Prerequisites	None	Co-requisites	None		
Course Content	<p><u>Introduction to Computer Science</u></p> <p>The Computer Scientist:</p>				

Describe the main Computer Science fields of study and related careers. Understand what is needed by a Computer Science student in order to become a professional Computer Scientist. Understand the ethical responsibilities of Computer Scientists. Comprehend the difference between a computer scientist and a computer programmer, software engineer, information scientist, etc.

Computer Science concepts (information representation):

Understand the main information representation methodologies used in modern computing systems. Understand how binary values are used to represent numbers, text, audio and video information.

Computer Science concepts (hardware):

Learn about the basic low-level and high-level components employed for the implementation of computing functions. Logic gates, Von Neumann architecture, storage devices.

Computer Science concepts (areas)

Discuss the areas of computer science and explain how they inter-operate to produce results. Briefly introduce: programming, algorithms, operating systems, databases and information systems. software engineering, data communication and networking.

Programming principles

Computer Science concepts (programming):

Understand the concepts of programming and programming languages. Define the difference between interpretation and compilation. Understand the difference between low-level and high-level programming languages.

Introduction To Programming: Primitive data types and variable definition. Basic input output statements, formatting program output, arithmetic operators, compound assignment operators, order of precedence. Writing, compiling and executing a simple computer program.

Conditionals, logical operators, Control Structures: <if> <if/else> <switch>. Repetition Structures: <for> <while>; the break and continue statements.

Functional abstraction. Function definitions and prototypes, arguments and parameter passing. Scope and duration of variables. Passing by value or reference. Returning values from functions. Argument promotion and casting. Function overloading.

Arrays: Introduction to arrays, array declaration and allocation, initializing array elements, accessing array elements, manipulation of arrays, passing arrays to functions. Characters & strings, string manipulation functions, variable size strings.

	<p>Pointers (if applicable): Pointer variables, working with pointers, pointer operators and pointer arithmetic. Calling functions by reference. Using pointers in place of arrays. Pointers to pointers, 2D arrays using pointers. Arrays of pointers; pointers to functions.</p> <p>Problem Solving: Representing and refining algorithms, using Sub-programs for sub-problems, decision steps in algorithms, tracing a program on algorithm, problem solving strategies, generalizing a solution, debugging and testing programs, common programming errors.</p> <p><u>Object Orientation</u></p> <p>Introduction to objects and object-oriented design, classes, constructors, overloaded constructors, get and set methods, class-wide variables, class scope, data abstraction and encapsulation. Class composition</p> <p>Inheritance and Polymorphism: super/base classes and sub/derived classes, inheritance, polymorphism, overriding. Creating hierarchies of classes and using them to describe real life problems/situations.</p>								
Teaching Methodology	Face-to-Face								
Bibliography	<ul style="list-style-type: none"> • Dale N. and Lewis J.; COMPUTER SCIENCE ILLUMINATED; Jones & Bartlett Learning • Deitel P.J and Deitel H.; C, HOW TO PROGRAM; Pearson • Deitel and Deitel; C++ HOW TO PROGRAM; Pearson • Deitel and Deitel: JAVA HOW TO PROGRAM; Pearson • Deitel and Deitel: INTRO TO PYTHON FOR COMPUTER SCIENCE AND DATA SCIENCE, LEARNING TO PROGRAM WITH AI, BIG DATA AND THE CLOUD, Pearson 								
Assessment	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding: 5px;">Coursework</td> <td style="text-align: center; padding: 5px;">30%</td> </tr> <tr> <td style="padding: 5px;">Examinations</td> <td style="text-align: center; padding: 5px;">60%</td> </tr> <tr> <td style="padding: 5px;">Class participation and Attendance</td> <td style="text-align: center; padding: 5px;">10%</td> </tr> <tr> <td style="padding: 5px;">Total</td> <td style="text-align: center; padding: 5px;">100%</td> </tr> </table>	Coursework	30%	Examinations	60%	Class participation and Attendance	10%	Total	100%
Coursework	30%								
Examinations	60%								
Class participation and Attendance	10%								
Total	100%								
Language	English								