Course Title	Programming Principles II - Robotics lab				
Course Code	CSE120				
Course Type	Compulsory				
Level	Bachelor (1st cycle)				
Year / Semester	1st year / 2nd semester				
Teacher's Name	TBA				
ECTS	6	Lectures / week	3 hours/14 weeks	Laboratories / week	None
Course Purpose and Objectives	The purpose of this course is to introduce basic principles of object orientation and have the students develop an understanding of concepts and ideas relating to object orientation. An objective of the course is to have student create and learn to use classes in order to solve problems by applying the object orientation paradigm. Another objective is to have the student learn to use a high-level object-oriented programming language as a means to providing solutions. Another objective of the course is to increase the programming experience of the students and reinforce their knowledge of a basic computer science skill.				
Learning Outcomes	Upon successful completion of this course students should be able to • specify new classes and provide the basic building blocks (constructors functions, get functions, set functions etc.) • create objects of specified classes and use them to design and implement solutions to scenario based problems • use inheritance and implement hierarchies of classes and dynamic object orientated features • work with string and string manipulation functions • write to and read from files • apply error checking techniques • depending on the programming language used: a. declare and use pointers and/or b. create abstract classes and interfaces and/or c. create basic graphical user interfaces and handle events and/or d. specify and use operator overloading				
Prerequisites	CSE100		Co-requisites	None	
Course Content	Classes and objects Introduction to classes and object-oriented design; creating classes; preprocessor directives and integrating classes in the native language environment; private, protected and public acces; writing constructors (overloading constructors or providing default argument constructors); access (set & get) functions; static class members;				

	constant class members; data abstraction and encapsulation. Printing or outputting objects Creating objects; using objects Object creation, arrays of objects, objects as arguments to functions, objects as return values form functions, pointers to objects, object references; using objects as data attributes of other classes (class composition); writing additional class functions and defining additional object behaviors. Inheritance and polymorphism The concept of inheritance and building classes based on (or inheriting form) other, existing, classes. Understanding code reusability and applying it in creating new classes based on existing ones. Depending on the programming language used: ▪ Base classes and derived classes or super classes and subclasses ▪ Abstract classes or virtual classes ▪ Interfaces or multiple inheritance ▪ Polymorphism and dynamic binding Exceptions and exception handling Errors, types of errors, error checking approaches; error handling approaches; the notion of exception and what it means to either throw one or catch one. securing programs to ensure correct and uninterrupted program execution. Strings and string manipulation Usual data types; using strings; manipulating strings, comparing, changing, truncating, concatenating strings. Files: Temporary vs permanent storage, what is a file; types of files; saving data onto files and retrieving data from files. Ready-made (programming language-specific) functions to output data to files or input data from files. Writing data as text or as binary. Depending on the programming language used: ▪ Friend functions and Friend classes and/or ▪ Operator overloading and/or ▪ Graphical user interface; GUI components; building GUI applications and/or ▪ Event handling, inner classes
Teaching Methodology	Face-to-Face
Bibliography	Deitel P., Deitel H., C++ How to program: Late objects Pearson Latest edition Deitel P., Deitel H., Java How to program: Late objects Pearson

<table>
<tr><td></td><td>Latest edition

Savitch W., Mock K.,
Absolute C++
Pearson
Latest edition

Savitch W., Mock K.,
Absolute Java
Pearson
Latest edition

Stroustrup B.,
Programming: Principles and practice using C++
Addison-Wesley Professional
Latest edition

Sedgewick R., Wayne K.,
Introduction to programming in java: An Interdisciplinary approach
Addison-Wesley Professional
Latest Eduction</td></tr>
<tr><td>Assessment</td><td>Class Participation and attendance 10%
Coursework 30%
Midterm examination 30%
Final examination 30%
100%</td></tr>
<tr><td>Language</td><td>English</td></tr>
</table>