



Computer Programming Technology

Syllabus

Course code:	6078
Number of ECTS credits:	6
Semester:	1st (September-December)
Prerequisites:	None
Recommended components:	Programming Fundamentals (6074)
Language of instruction:	Spanish

Course description

This course explains the main elements of the Object Oriented Programming (OOP) paradigm and it introduces Abstract Data Type (ADT) as basic concept for object-orientation. This course is motivated from the perspective of the need to solve complex problems and the limitations that can be found when trying to use just the structured programming paradigm as explained in the *Programming Fundamentals* course. Different concepts such as classes, objects, abstraction, encapsulation, polymorphism, inheritance, linear ADT and non-linear ADTs are the main aspects being considered. Python is the programming language being used.

Learning outcomes and competences

1. Creating mathematical documents where programming problems are solved.
2. Using classes and objects as part of solving a problem.
3. Knowing the types of abstract data and some implementation techniques.
4. Correctly applying access modifiers: public, private, protected.
5. Being able to differentiate between the concept of inheritance and interface.
6. Invoking object methods and class methods
7. Applying method overloading and overwriting.
8. Managing exceptions.
9. Managing the input/output interfaces

Course contents

I. Object Oriented Programming

1. Classes and objects.
Introduction, need and basic concepts. The life cycle of an object: definition, declaration, construction, use and destruction.
2. Members and visibility
Members of a class. Types of members. Namespace. Access modifiers.
3. Hiding and overloading.
Access control mechanisms to objects. Hiding at the level of package and class. Overloading constructors and methods.
4. Hierarchy
Types of relationships between classes. Customer relationship between classes (composition vs. aggregation). Inheritance and types. Overwriting and hiding. Polymorphism. Composition vs inheritance.
5. Interfaces
Abstract classes. Informal interfaces vs formal interfaces. Polymorphism. Inheritance and interfaces. Loss and recovery of type.
6. Input/Output and Exceptions
Streams. Files. Serialization. Exception control. Inheritance and exceptions.
7. Graphical User Interface
Observer and MVC patterns. GUI construction fundamentals.

II. Abstract Data Type

1. Abstract data types
Abstraction and types. Abstract Data Types (ADTs). Specification, representation and implementation of ADTs.
2. Linear ADTs
Arrays. Lists. Stacks. Queues Vectors. Sets. Dictionaries.
3. Non Linear ADTs
Trees. Graphs.

References

Main texts

1. Lutz, Mark. *Learning Python*. Sebastopol, CA : O'Reilly. (2013)
2. Rance D. Necaie. *Data Structures and Algorithms using Python*. John Wiley and Sons. 2011
3. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser. *Data Structures and Algorithms in Python*. Wiley. John Wiley and Sons. 2013
4. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser. *Data Structures and Algorithms in Java*. Sixth Edition. Wiley. John Wiley and Sons. 2014