Object-Oriented Programming (COMP0004)

Description

Aims:
This module provides a thorough introduction to object-oriented (OO) programming. The key object-oriented concepts such as class design, inheritance, dynamic binding and interfaces, are introduced, along with design strategies for smaller sized object-oriented programs. Material is also included on data representation and data processing, the implementation of common data structures such as lists, trees and hash tables, basic cloud-based applications, and on development tools and practices. To learn how to apply object-oriented design concepts, and practice programming skills, the module has a substantial practical element, with weekly lab classes, sets of exercise questions, and an individual programming project.

Learning outcomes:
On successful completion of the module, a student will be able to:
1. Design and implement smaller-scale object-oriented programs;
2. Have a good knowledge of key object-oriented principles and design strategies;
3. Read, write, and process data sets using object-oriented techniques;
4. Understand how core data structures are implemented using classes;
5. Designing and running basic OO applications in the cloud;
6. Use programming tools such as an integrated development environment (IDE), debugger, and version control;

Content:
Topics covered in lectures or via online material:

Object-Oriented Principles:
- Taking an object-oriented approach to software development;

Key information

Year: 2019/20
Credit value: 15 (150 study hours)
Delivery: UG L4, Campus-based
Reading List: View on UCL website
Tutor: Dr Graham Roberts
Term: Term 2
Timetable: View on UCL website

Assessment

- Written examination (departmentally managed): 20%
- Coursework: 80%

Find out more

For more information about the department, programmes, relevant open days and to browse other modules, visit ucl.ac.uk
- Classes and the use of abstractions;
- Key design principles and development practices;

Object-Oriented Programming:
- Designing and implementing classes;
- Identifying and implementing class relationships;
- Using abstract classes, interfaces and inheritance;
- Developing smaller-scale programs taking an object-oriented approach;
- Effective object-oriented design and programming practices;
- Documenting designs using class diagrams;
- The design of basic cloud-based applications;

Data Representation:
- Implementing common data structures;
- Storing data in files and basic databases;
- Processing data using streams;

Requisites:
In order to be eligible to select this module, a student must be registered on a programme for which it is a formally-approved option or elective choice AND must have taken Principles of Programming (COMP0002) in Term 1.