

## MODULE DESCRIPTOR

<b>MODULE TITLE</b>	Introduction to Programming		
<b>MODULE CODE</b>	CO1407 (L4)	<b>CREDIT VALUE</b>	20 credits (10 ECTS)
<b>SCHOOL</b>	SCHOOL OF SCIENCE		

### MODULE AIMS

1. To develop basic skills in coding, editing, compiling, debugging and testing computer programs.
2. To develop skills in analysing problems and designing software solutions.

### MODULE CONTENT

In this module students learn how to write simple programs. No prior programming experience is assumed. To provide context and motivation, examples and practical exercises relate loosely to the different courses that the students may progress onto.

The module uses a single programming language and programming environment, but develops transferable programming skills by focusing on the fundamental concepts used in the majority of programming languages. From the outset, attention will be paid to program readability and maintainability.

#### Implementation

- Overall program structure and program flow
- Variables and data types
- Assignment and simple operators
- Simple I/O
- Conditions (if/switch statements)
- Iteration (for and while loops)
- Functions and parameters, parameter passing mechanisms
- Input / output, including file I/O
- Structures
- Arrays
- Readability and maintainability of program code (e.g. layout, comments, constants)

#### Use of a Programming Environment

- Editor, compiler, debugger and on-line help

#### Program Testing and Debugging

### INTENDED LEARNING OUTCOMES

On successful completion of this module a student will be able to:

1. Design an appropriate solution for a given programming problem.
2. Implement a readable and maintainable software solution based on his or her own design.
3. Develop an appropriate test strategy and apply it to a software solution.
4. Evaluate the quality of his or her developed software

### TEACHING METHODS

Lectures will be used to introduce concepts, methods and programming techniques.

Practical classes will be used primarily to develop individual skills and competences in the use of the range of facilities offered by the programming environment in the development, testing and debugging of program solutions.

---

Practical sessions present a range of intermediate problems that require the use of the concepts introduced in the lectures. Exercises each week begin simple (e.g. add minor functionality to a given, working program) and increase in difficulty until the student is writing modest programs from scratch. Advanced exercises are provided to stretch those students who adapt quickly to the module.

Practical exercise problems will be linked to lecture classes but the student will be expected to prepare solutions to exercises in advance of the practical class.

Lectures or practical sessions may take the form of guided workshops / demonstrations. These will be used to practically demonstrate key programming tasks and tools.

Drop-in support sessions provide extra guidance to help students progress with the practical material, and also to enthuse those who are interested in taking the exercises further. In addition, there will be guided workshops on the use of a programming environment and on debugging programs.

Assessment consists of:

- A practical assignment, which will reinforce and extend the work covered in practical classes.
- An exam to test the students understanding of the theory presented in the module.

---

## **ASSESSMENT METHODS**

This module is assessed through a Practical Assignment based on lab sheets (60%) and a timed examination (40%).