

| <i>Pre-Requisite Modules code(s)</i> | <i>Co-Requisite Modules code(s)</i> | ECTS Credits | Module Code | Programming and Algorithms 1 |
|--------------------------------------|-------------------------------------|--------------|-------------|-------------------------------------|
| | | 10 | | |

8.1.5 Programming and Algorithms 1

Module Author

Programming and Algorithms Group

Module Description

This module is an introduction to programming, program design and algorithms. Students are introduced to a structured procedural programming language and to commonly used algorithms and their implementation.

This module assumes no prior knowledge of programming or algorithms.

Module Aims

The aims of this module are to:

- Teach the fundamentals of procedural programming
- Teach the principles of good program design, implementation, documentation and testing.
- Teach the theory and application of elementary algorithms and data structures.

Learning Outcomes

On completion of this module, the successful learner will be able to:

1. Design and write computer elementary programs in a structured procedural language.
2. Use a text editor with command line tools and simple Integrated Development Environment (IDE) to compile, link and execute program code.
3. Divide a computer program into modules.
4. Test computer programs to ensure compliance with requirements.
5. Implement elementary algorithms and data structures in a procedural language.

Learning and Teaching Methods

Lectures, self-directed learning, practical laboratory exercises, tutorials, problem-solving exercises.

A Virtual Learning Environment (VLE) is used extensively in this module.

| <i>Pre-Requisite Modules code(s)</i> | <i>Co-Requisite Modules code(s)</i> | ECTS Credits | Module Code | Programming and Algorithms 1 |
|--------------------------------------|-------------------------------------|--------------|-------------|-------------------------------------|
| | | 10 | | |

Module content

Introduction: What is a program? Source code. Machine code. Editing, Compiling, Linking Debugging. Use of an Integrated Development Environment (IDE).

Basic Data Types: integer, floating-point and character data and variables.

Basic Input-Output: Display data on a screen. Input data from the keyboard.

Programming Structures: Conditional statements: Boolean values and expressions, logical and relational operators, if-statement, case-statement, compound conditional statements.

Iterative constructs: while-statements, for-statements and nested control statements.

Structured Programming: functions, parameter passing, returning values.

Introduction to Data Structures: single-dimensional arrays, two-dimensional arrays, dynamically allocated arrays,

Basic Algorithms: summation, counting, numeric operations, swapping, maximum and minimum, simple array manipulation.

Elementary Sorting Algorithms: Internal Sorting. Exchange sort. Interchange sort. Bubble sort. Shaker sort. Insertion sort.

Testing and debugging: Objectives and principles of testing. Choosing appropriate test data. Simple debugging using a program trace.

Documentation: Style guidelines.

Module Assessment

Assessment of the module will be as follows:

Continuous Assessment (30%):

- Individual assignments
- On-line tests
- In-class written tests

Written examination(70%):

- One two hour, end of module examination.

Essential Reading

Depending on the procedural language used in this module, specific reading lists will be specified in advanced of the start of the module.

| <i>Pre-Requisite Modules code(s)</i> | Co- Requisite Modules code(s) | ECTS Credits | Module Code | Programming and Algorithms 1 |
|--|--|-----------------|----------------|-------------------------------------|
| | | 10 | | |

Supplemental Reading

Robert Sedgewick (1997), Algorithms in C, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching (3rd Edition), Addison-Wesley