

<b>Pre-Requisite Modules code(s)</b>	<b>Co-Requisite Modules code(s)</b>	<b>ECTS Credits</b>	<b>Module Code</b>	<b>Module Title</b>
<b>None</b>	<b>None</b>	<b>5</b>	<b>CMPU1024</b>	<b>Program Design</b>

### 8.1.4. Program Design

**Module authors: Richard Lawlor and Shane Mulligan.**

#### Module Description

This module is concerned with program design skills, with particular reference to using flowcharts, pseudocode and programming language constructs to model and design computer programs. Consideration is given as to how problem information might be represented in code or on paper and what program design steps may be performed to arrive at a solution. Abstraction, modularity and top-down design are central to this module.

#### Module Aims

- To introduce students to program design and the main techniques of program design. In particular, to introduce students to design strategies such as top-down and bottom-up and the techniques of stepwise refinement.
- To introduce the use of pseudocode and flowcharts in program design.
- To convey to students an understanding and appreciation of the power of abstraction whereby the essential information relating to a programming problem is abstracted and mapped onto programming constructs.
- To convey the importance of a well conceived design before rushing into code.

#### Learning Outcomes

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

- Abstract problem information and represent it on paper or an appropriate computing environment.
- Demonstrate a basic competence in the use of a program constructs to solve a problem
- Develop solutions to some elementary program design problems using top down design and stepwise refinement.
- Describe some simple program designs using pseudocode and flowcharts, and then implement the design.

#### Learning and Teaching Methods

The module will be delivered primarily through lectures, tutorials and laboratory work. The material will be developed in an informal way during lectures. It is envisaged that the students will assimilate much of the material through problem solving and exercises. Emphasis will be placed on worked examples and group discussion of exercises.

Practical work will consist of weekly laboratory sessions. This will also help the students understand how program design concepts can be mapped to a program language constructs.

<b>Pre-Requisite Modules code(s)</b>	<b>Co-Requisite Modules code(s)</b>	<b>ECTS Credits</b>	<b>Module Code</b>	<b>Module Title</b>
<b>None</b>	<b>None</b>	<b>5</b>	<b>CMPU1024</b>	<b>Program Design</b>

A large emphasis will be placed on allowing the students to tryout the problem descriptions and possible solutions in the laboratory. Exercises will be provided that cover material relevant to the implementation of problem solutions. These exercises get progressively more difficult and will incorporate material learned previously.

Tutorials will be used to allow the students to get help in the more difficult areas and provide students with the opportunity for individual assistance from the supervisor.

### Module Content

- Problem Solving, Stages in Problem Solving.
- Data Types and Data Representation.
- Program Constructs.
- Abstraction, Problem Specification, Approaches to Problem Solving and Program Construction, Divide and Conquer, Stepwise Refinement, Top Down Design, Bottom Up Design.
- Recursion. Greatest Common Divisor, Factorial and Fibonacci.
- Pseudocode and Flowcharts in Program Design.
- Linear Data Structures - arrays and lists.

### Module Assessment

Assessment will be based on a two hour end of semester written exam and continuous assessment during the semester.

Written exam - 60%

Continuous Assessment - 40%

### Essential Reading

No specific textbook. Lecture notes and laboratory material as provided by the lecturer.

### Supplemental Reading

Maureen Sprankle and Jim Hubbard - Problem Solving and Programming Concepts.

### Web References

As specified by the lecturer.

### Further Details

One semester (semester 1) module: 2 lectures per week, 1 laboratory hour, 1 tutorial hour

**Date of Academic Council approval .....**