

<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>CMPU1014</b>	<b>Introduction to Algorithms</b>

### 8.1.7. Introduction to Algorithms

**Module authors: Shane Mulligan.**

#### Module Description

This module is concerned with algorithms and data structures, and their implementation in a suitable language. A range of common data structures will be studied, and the algorithms that use these data structures. Students will gain a deeper understanding and competence in the algorithms studied, through their implementation, testing and comparison.

#### Module Aims

The aim of this module is:

- To introduce the student to a variety of data structures and how they may be implemented.
- To present the student with algorithms that make use of the data structures studied.
- To study the algorithms through the use of paper-based examples.
- To represent algorithms by the use of pseudo-code and/or flowcharts.
- To implement a selection of data structures and algorithms, test them and compare them.

#### Learning Outcomes

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

- Define a variety of data structures and their associated operations.
- Demonstrate a knowledge of standard algorithms by expressing them using pseudo-code.
- Analyse an algorithm in order to compute its complexity/operation count.
- Implement a range of data structures and algorithms, and test them to increase their knowledge and competence.

#### Learning and Teaching Methods

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

Practical work will consist of weekly laboratory sessions. This will also help the students understand how data structures and algorithms can be implemented in a programming language.

An emphasis will be placed on allowing the students to experiment with their implementations, e.g. by changing the input data.

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.

<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>CMPU1014</b>	<b>Introduction to Algorithms</b>

## Module Content

- Algorithms, Properties.
- Data Structures, Abstract Data Types.
- Common Data Structures, Linked-lists, Stacks, Queues, Trees.
- Standard algorithms, searching and sorting.
- Analysis of algorithms, computational complexity.
- Implementation and testing of algorithms.

## Module Assessment

Assessment will be based on a two hour end of semester written exam and continuous assessment during the semester. The continuous assessment may consist of laboratory tests, practical assignments or a combination of these.

Written exam - 60%

Continuous Assessment - 40%

## Essential Reading

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

## Supplemental Reading

Deitel and Deitel – 2007, C How to program, 5<sup>th</sup> Edition, Prentice Hall.

## Web References

As specified by the lecturer.

## Further Details

One semester (semester 1) module and each week:

2 lectures per week

1 laboratory hour

1 tutorial hour

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