CO1107 Algorithms, Data Structures and Advanced Programming

Credits: *15* **Convenor:** *Dr. T. Ridge* **Semester:** *2nd*

Prerequisites:	*Essential: CO1101, CO1102, CO1105* *Desirable: -*
Lectures:	*22 hours*
Tutorials:	*6 hours*
Laboratories:	*20 hours*
Independent Study:	*102 hours*
Assessment:	*Coursework: 100% + Exam: 0%*

Formative Coursework	*Summative Coursework*	
None	**Class Tests:** *3*	*in total*
	Assignments: *0*	*in total*

Learning Outcomes **Students should be able to:**

- Explain and critique high level overviews of programming paradigms;
- Define and explain advanced datatypes such as stacks, queues, lists, trees and graphs; write simple programs;
- Define, discuss and explain the main algorithms and techniques (such as sorting, searching, hashing, traversal and recursion) and write programs using these algorithms;
- Solve problems by designing programs in a range of advanced topics such as XML tools, threading, sockets, GUIs and elementary games.

Explanation of Prerequisites

A good knowledge of Python and basic programming is essential before taking this module.

Module Description

This module covers advanced programming material, including common algorithms and data structures.

Syllabus

Review of basic programming. Procedural programming. Basic object-oriented programming in Python. (Revision of previous modules.)

Review of essential datatypes: sets and maps.

Algorithms and data structures. What is an algorithm? What is a data structure?

Standard list-like data structures (lists, queues, stacks). Simple functions over list-like data (e.g. reverse a list).

Advanced programming technique: recursion. Basic recursion over natural numbers (e.g. calculation of Fibonacci numbers). Structural recursion over lists.

Sorting and searching. Sequential search. Binary search. Bubble sort. Insertion sort. Merge sort.

Trees. Examples of trees. Tree vocabulary. Tree implementation in Python. Tree traversal (in-order, pre-order, post-order).

Graphs. Graph implementation in Python. Breadth-first search. Dijkstra's shortest path algorithm.

Further topics: JSON

Reading List

Runestone Academy have two excellent online courses. The first is "How to Think Like a Computer Scientist: Interactive Edition". This covers basic Python that you should have learnt in previous modules.

The second course is much more closely related to this module. It is called "Problem Solving with Algorithms and Data Structures using Python". Both courses are free. Sign up at the following address:

`https://runestone.academy/runestone`

The main Python website has many tutorials on Python. In addition, there are many books on Python available from the library (either in physical form, or as an e-book). However, there are relatively few books on data structures and algorithms in Python. For this reason, I recommend the online material above.