WM908-15 Programming and Fundamental Algorithms

23/24

Department
> WMG

Level
> Taught Postgraduate Level

Module leader
> Kurt Debattista

Credit value
> 15

Module duration
> 2 weeks

Assessment
> Multiple

Study location
> University of Warwick main campus, Coventry

Description

Introductory description

Current progress in information technology has meant the majority of organisations are using IT to innovate, and without an understanding of fundamental computing concepts future managers, scientists, engineers will be unable to take strategical decisions and provide critical thinking on most projects in full confidence.

The module is focused around two core themes in computing: algorithms and programming/development. These two concepts go hand in hand and it is understood that to be a good developer, algorithmic concepts need to be comprehensively understood and students should be able to critical apply knowledge from the theoretical aspects towards practical implementations of solutions for complex system designs in business, engineering, science and IT.

The programming focus is based on a high level programming language such as the C/C++ programming language and/or python, considered by many as one of the more useful programming languages and still the language of choice in many industries – once mastered the transition to the other popular programming languages Java, C-sharp, Objective C can be relatively straightforward. Importantly, this is not about teaching programming but about forming a conceptual understanding of computing principles with programming as a vehicle to further grasp

these concepts with the added bonus of adding an important skill to the CV. Future editions may adopt other programming languages.

Module aims

The over-arching aim of this module is: Give students the ability and confidence to solve problems efficiently using computers.

The sub-aims of this module are: Form an understanding of some classic algorithms from the literature Develop the understanding of which solutions/algorithmic paradigms work best for certain types of problems Design straightforward algorithms for yet unseen problems that have straightforward solutions Learn programming methods and how to design good code for a proposed algorithm.

Outline syllabus

This is an indicative module outline only to give an indication of the sort of topics that may be covered. Actual sessions held may differ.

Introduction Sorting as an introduction to algorithms
Data structures
Complexity and decision making
Brute force and divide and conquer methods for solving problems
Dynamic programmig and greedy methods
Exhaustive search and recursion
Advanced Data structures
Graph algorithms and data structure– algorithms based on graph theory for solving problems that can be expressed as graphs
Misc. algorithms (eg string matching, spatial data structures etc.) as part of in-class tutorials, introduced throughout the module.
Limitations of algorithms and coping with limitations
Conclusions, recap and next steps

Tutorials
Introduction to programming
Introduction to Types and Commands
Dealing with pointers
Generics and abstract data types
Concepts of Object Oriented Programming Inheritance Polymorphism
File I/O
Introduction to multithreading

4 x problem solving examples in class (

Demonstrations and group work Worksheets solving one/two problems from each lecture 2 versions of problems one more advanced for more advanced students Group project (in class) – learn advantages, pitfalls and practicalities of programming as part of a team

Learning outcomes

By the end of the module, students should be able to:

- Master a sound, conceptual understanding of the theory and concepts of programming and fundamental algorithms and data structures.
- Autonomously distinguish the right solution for a given problem from amongst a set of algorithmic and programming tools.
- Program effectively and independently in a high level programming language at an intermediate level.
- Take, straightforward to complex, algorithmic concepts, whether created or based on literature and implement them correctly.

Research element

Many research students have attended the 10 CATS version of this course before (and particularly in its previous incarnation as CY903) - it forms a sold foundation for students working in the mathematical sciences to learn how to program and run code. Many of the in class examples are, usually, simulation based.

Interdisciplinary

The examples shown in terms of programming will cover a number of areas around science and engineering, broadly falling in the mathematical sciences domain.

Subject specific skills

Mathematical skills, programming skills

Transferable skills

Technology literacy, adaptability

Study

Study time

Type	Required
Lectures	10 sessions of 1 hour 30 minutes (10%)
Seminars	1 session of 1 hour 30 minutes (1%)
Tutorials	9 sessions of 1 hour 30 minutes (9%)
Online learning (independent)	7 sessions of 1 hour 30 minutes (7%)
Other activity	29 hours 30 minutes (19%)
Assessment	80 hours (53%)
Total	150 hours

Private study description

No private study requirements defined for this module.

Other activity description

29 hours of student self directed study in preperation for both IMA and in class work. Guidance on self directed study will be provided in class.

Costs

No further costs have been identified for this module.

Assessment

You must pass all assessment components to pass the module.

Assessment group A1

	Weighting	Study time
PMA Project	100%	80 hours

This is the post module assignment. It is an original programming application with a brief report outlining motivation and solutions.

Assessment group R1

	Weighting	Study time
PMA Project	100%	

This is the post module assignment. It is an original programming application with a brief report outlining motivation and solutions.

Feedback on assessment

The Take Home assignment will be marked and feedback provided before the modules finishes. Feedback will also be provided in class via the tutor demonstrating a solution to the same problem.

Availability

Pre-requisites

For students outside of SCAV, CSM and SAE tutor must be consulted to see if student is suitable.

Post-requisite modules

If you pass this module, you can take:

- WM9M2-15 Computer Graphics

There is currently no information about the courses for which this module is core or optional.