

## Liverpool John Moores University

Title: MATHEMATICAL COMPUTER PROGRAMMING  
Status: Definitive  
Code: **4112MATHS** (124192)  
Version Start Date: 01-08-2021

Owning School/Faculty: Computer Science and Mathematics  
Teaching School/Faculty: Computer Science and Mathematics

Team	Leader
Ross Kelly	Y

**Academic Level:** FHEQ4      **Credit Value:** 20      **Total Delivered Hours:** 55  
**Total Learning Hours:** 200      **Private Study:** 145

### Delivery Options

Course typically offered: Semester 1

Component	Contact Hours
Lecture	22
Practical	33

**Grading Basis:** 40 %

### Assessment Details

Category	Short Description	Description	Weighting (%)	Exam Duration
Report	AS1	Programming tasks	40	
Report	AS2	Software development	60	

### Aims

*To develop IT problem solving skills*  
*To become familiar with a range of mathematical programming techniques*  
*To gain an understanding of how software is developed*  
*To prepare students for mathematical software development at higher levels, both work and study*

## Learning Outcomes

After completing the module the student should be able to:

- 1 Apply knowledge of computer programming constructs and algorithms to IT problems.
- 2 Demonstrate problem solving skills to create simple software solutions.
- 3 Evaluate alternatives and make sound judgements about data structures.
- 4 Investigate development environment tools for use in software development.
- 5 Demonstrate familiarity with using mathematical functions within programs.

## Learning Outcomes of Assessments

The assessment item list is assessed via the learning outcomes listed:

Coursework 1	1	2		
Coursework 2	3	4	5	

## Outline Syllabus

### *Computers and Computer Programming*

- How programs work within computers
- Current programming languages and their evolution

### *Integrated Development Environment*

- Working with code
- Compiling, profiling, testing and organising code

### *Basic elements of programs*

- Syntax
- Variables/Types
- Expressions
- Input/Output and Devices
- Classes and methods

### *Control structures*

- Conditionals / selection
- Loops / repetition
- Logical problem solving
- User defined classes
- Value and Reference Types
- Arrays / Collections
- String manipulation
- Code structure, procedures/methods, callbacks.
- Recursion

### *Graphics*

- Plotting graphs and statistical data

## **Learning Activities**

Lectures – to introduce the programming theories and techniques

Lab exercises – programs for students to write and test.

Further exercises – practical examples for students to work on in their own time

Directed reading – background reading to enable the lab work to be completed.

## **Notes**

This module aims to develop programming and problem solving skills in students to help prepare them for work in mathematics and statistics.