

## Liverpool John Moores University

Title: PROBLEM SOLVING FOR INFORMATION SYSTEMS  
Status: Definitive  
Code: **4111COMP** (121209)  
Version Start Date: 01-08-2021

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

| Team         | Leader |
|--------------|--------|
| Denis Reilly | Y      |
| Mark Allen   |        |

**Academic Level:** FHEQ4      **Credit Value:** 20      **Total Delivered Hours:** 55

**Total Learning Hours:** 200      **Private Study:** 145

### Delivery Options

Course typically offered: Semester 2

| Component | Contact Hours |
|-----------|---------------|
| Lecture   | 11            |
| Practical | 44            |

**Grading Basis:** 40 %

### Assessment Details

| Category   | Short Description | Description             | Weighting (%) | Exam Duration |
|------------|-------------------|-------------------------|---------------|---------------|
| Report     | AS1               | Design Model            | 40            |               |
| Technology | AS2               | Software Implementation | 60            |               |

### Aims

*To develop skills in computational thinking that can be used to develop programs to solve subject specific problems*

### Learning Outcomes

After completing the module the student should be able to:

- 1 Use computational thinking to design solutions to problems
- 2 Implement design solutions in a suitable programming language
- 3 Develop test plans

## Learning Outcomes of Assessments

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

|                         |   |   |
|-------------------------|---|---|
| Design Model            | 1 |   |
| Software Implementation | 2 | 3 |

## Outline Syllabus

*Computational thinking – decomposition, pattern recognition, abstraction, algorithms*  
*Top-down design/successive refinement*  
*Pseudo-code and diagram techniques*  
*Practical exercises using compound control structure*  
*Practical exercises using methods/functions*  
*Solving problems with classes/objects*  
*Practical exercises using classes/objects*  
*Practical exercises on error handling*  
*Testing – test data, test cases, test plans, test strategies (unit, system)*  
*Practical exercises on testing*

## Learning Activities

Learning will largely be based on practical exercises and problem solving activities. Lectures will be used to introduce topics, which will be reinforced through practical work.

## Notes

The practical exercises will be related such that they lead to the development of a larger software implementation.