

Liverpool John Moores University

Title: Software Engineering Workshop
Status: Definitive
Code: **4221COMP** (127977)
Version Start Date: 01-08-2021

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

Team	Leader
David Lamb	Y
Hoshang Kolivand	

Academic Level: FHEQ4 **Credit Value:** 20 **Total Delivered Hours:** 44
Total Learning Hours: 200 **Private Study:** 156

Delivery Options

Course typically offered: Semester 2

Component	Contact Hours
Workshop	44

Grading Basis: 40 %

Assessment Details

Category	Short Description	Description	Weighting (%)	Exam Duration
Artefacts	AS1	Group development of software using Agile methods	100	

Aims

To introduce the students to practical, team-based software design, development and evaluation.

To develop the students' programming skills with a considerable increase in programme complexity.

To become familiar with and utilise appropriate professional Software Engineering skills relating to project planning, team and client communication, design documentation, along with versioning and management of their software source and

binaries.

Learning Outcomes

After completing the module the student should be able to:

- 1 Plan and manage a team-based software development project, utilising appropriate incremental software development lifecycle methodologies.
- 2 Design and document software system development using industry-standard techniques.
- 3 Evaluate the quality of software design and implementation; refactor extant software.
- 4 Become familiar with modern development ecosystems; such as integrated development environments and source code management repositories.

Learning Outcomes of Assessments

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

Group development of software	1	2	3	4
-------------------------------	---	---	---	---

Outline Syllabus

Group Planning and Management

Software Development Lifecycles

Object-Orientation Basics: Composite data types and design documentation standards

Source code management and versioning

Source documentation practices

Test strategies; User/Acceptance, TDD and Unit Testing

Debugging and state inspection

Design Patterns Introduction

Refactoring

Profiling: evaluating the performance of executing code

Learning Activities

Brief didactic learning activities will be based around a given problem topic, from the outline syllabus. However, the primary learning technique in workshop sessions will be problem-based learning. Students will form software development teams to develop software from specification through design and implementation.

Notes

This module is intended to give students a comprehensive introduction to applied software development on real-world problems, utilising team-working methods and industry-standard practices. In addition to the scheduled contact hours, teams will

need to utilise private study time to research and develop solutions. Teams will have considerable supported study activities to assist them in this – significantly, regular (student-led) team meetings with module staff to update progress and gain assistance. The module requires that students have received a basic introduction to programming in a higher-level programming language.