



## **Module Specification**

### **Advanced Software Development**

Version: 2024-25, v4.0, 05 Jul 2024

#### **Contents**

<b>Module Specification .....</b>	<b>1</b>
<b>Part 1: Information .....</b>	<b>2</b>
<b>Part 2: Description .....</b>	<b>2</b>
<b>Part 3: Teaching and learning methods .....</b>	<b>3</b>
<b>Part 4: Assessment.....</b>	<b>5</b>
<b>Part 5: Contributes towards .....</b>	<b>7</b>

## Part 1: Information

**Module title:** Advanced Software Development

**Module code:** UFCF8S-30-2

**Level:** Level 5

**For implementation from:** 2024-25

**UWE credit rating:** 30

**ECTS credit rating:** 15

**College:** College of Arts, Technology and Environment

**School:** CATE School of Computing and Creative Technologies

**Partner institutions:** None

**Field:** Computer Science and Creative Technologies

**Module type:** Module

**Pre-requisites:** Principles of Programming 2023-24

**Excluded combinations:** None

**Co-requisites:** None

**Continuing professional development:** No

**Professional, statutory or regulatory body requirements:** None

## Part 2: Description

**Overview:** This module teaches students how to apply contemporary software development approaches in order to develop advanced software applications such as distributed and concurrent software intensive database systems featuring GUI front ends.

**Features:** Not applicable

**Educational aims:** This module aims to build and the underlying skills developed at L4 to bring them together in the context of greater software development complexity.

**Outline syllabus:** Tools and techniques for problem analysis.

Software solution approaches and the requirements for a chosen software solution approach.

Approaches to the creation of high- and low-level designs for the chosen solution approach.

Software development methodologies such as Agile and the incremental approach.

Design representations and modelling.

Unified Modelling Language (UML).

Design principles: Design for resilience and system risk, the SOLID principles in object-oriented software development - for example, the single responsibility principle, open-closed principle etc.

Design patterns and their use.

Security by design principles.

Object-oriented programming language features to support threading and distributed systems.

Distributed databases.

User interface design.

Testing concepts and methods.

One Agile software development methodology will be taught in depth using, for example, Scrum or DevOps

Legal, ethical, social and professional issues.

Configuration management, software licensing, quality management, version control and contemporary emerging software development tools, technologies and methods.

### **Part 3: Teaching and learning methods**

**Teaching and learning methods:** In this module, the focus is on learning by doing. Through a combination of lectures and practical work, students will pursue a software development life cycle, from problem analysis through to implementation, testing and maintenance. Along the way, a range of methods, tools and techniques will be introduced and experimented with.

Throughout there is an emphasis on enquiry. Students will be required to justify their development choices to both their peers and to teaching staff. They will also be required to consider legal, ethical, social and professional implications of the activities undertaken. Group working will be used to mirror how software development is managed in industry.

**Module Learning outcomes:** On successful completion of this module students will achieve the following learning outcomes.

**MO1** Analyse problems in order to identify software-solution approaches and requirements for computer-based software-intensive systems.

**MO2** Compare and contrast software development methodologies and choose one suitable for a given application.

**MO3** Design, implement, test and manage reasonably sized software system considering database and GUI components.

**MO4** Discuss the need for security in the context of system development

**Hours to be allocated:** 300

**Contact hours:**

Independent study/self-guided study = 228 hours

Face-to-face learning = 72 hours

Total = 0

**Reading list:** The reading list for this module can be accessed at [readinglists.uwe.ac.uk](https://rl.talis.com/3/uwe/lists/939B5869-07ED-B7EC-90C9-B237896D16B9.html?lang=en-US) via the following link <https://rl.talis.com/3/uwe/lists/939B5869-07ED-B7EC-90C9-B237896D16B9.html?lang=en-US>

## Part 4: Assessment

**Assessment strategy:** The assessment is designed to ensure that students' understanding and skills are developed incrementally. The assessment strategy provides continual formative feedback opportunities through peer review and tutor progress reviews during computer practical sessions and allows students to develop their skills with the materials being presented in the lectures and practical sessions.

There are two assessment tasks for this module.

The first assessment task is a group portfolio assignment. It covers analysis of a problem, the specification of its associated requirements, the design, implementation and testing of a solution. Students are expected to select and apply a suitable agile software development methodology.

The second assessment task is a short viva and demo of the software. The demo/viva will provide students the opportunity to present and explain their work based on which the quality of their portfolio will be determined. The demo session will show quality of software features, explain the applied software methodology, clarify individual contributions and the overall evaluation of the project.

The mark for portfolio will be adjusted based on peer and tutor assessment of individual contributions to the group.

For the resit students may be reassigned to new groups to complete a group project. Like first sit, the resit group portfolio assignment task will be based on a related problem analysis, requirements specification, design, coding, testing and application of a suitable agile software development methodology. The demonstration task will be held online and should include demonstration of the software features, explanation and justification of the applied software methodology, individual contributions and overall evaluation of the project.

### **Assessment tasks:**

#### **Portfolio (First Sit)**

Description: The assessment of this task will be based on portfolio submission that includes requirements analysis, design, software code and testing evidence.

Weighting: 75 %

Final assessment: No

Group work: Yes

Learning outcomes tested: MO1, MO2, MO3, MO4

### **Presentation (First Sit)**

Description: Students are expected to take part in demo and viva session where students will demonstrate software features, explain and justify the software methodology, clarify individual contributions and critically evaluate the project and security measures. The quality of software implementation and suitability of agile approach will be assessed based on this task.

Weighting: 25 %

Final assessment: Yes

Group work: Yes

Learning outcomes tested: MO2, MO3, MO4

### **Portfolio (Resit)**

Description: The assessment of this task will be based on portfolio submission that includes requirements analysis, design, software code and testing evidence.

Weighting: 75 %

Final assessment: No

Group work: Yes

Learning outcomes tested: MO1, MO2, MO3, MO4

### **Presentation (Resit)**

Description: Students are expected to take part in demo and viva session where students will demonstrate software features, explain and justify the software methodology, clarify individual contributions and critically evaluate the project and security measures. The quality of software implementation and suitability of agile approach will be assessed based on this task.

Weighting: 25 %

Final assessment: Yes

Group work: Yes

Learning outcomes tested: MO2, MO3, MO4

## **Part 5: Contributes towards**

This module contributes towards the following programmes of study:

Computer Science [Villa] BSc (Hons) 2023-24

Computer Science [Villa] BSc (Hons) 2023-24

Computer Science (Artificial Intelligence) [NepalBrit] BSc (Hons) 2023-24

Computer Science (Artificial Intelligence) {Foundation} [GCET] BSc (Hons) 2022-23

Computer Science {Foundation} [Frenchay] BSc (Hons) 2022-23

Computer Science {Foundation} [Frenchay] BSc (Hons) 2022-23

Computer Science {Foundation} [GCET] BSc (Hons) 2022-23

Computer Science (Artificial Intelligence) {Foundation} [GCET] BSc (Hons) 2022-23

Computer Science [Frenchay] BSc (Hons) 2023-24

Computer Science [Frenchay] BSc (Hons) 2023-24

Computer Science [Phenikaa] BSc (Hons) 2023-24

Computer Science [Phenikaa] BSc (Hons) 2023-24