



## **Module Specification**

### **Advanced Software Development**

Version: 2023-24, v3.0, 16 Jan 2023

#### **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:** 2023-24

**UWE credit rating:** 30

**ECTS credit rating:** 15

**Faculty:** Faculty of Environment & Technology

**Department:** FET Dept of Computer Sci & Creative Tech

**Partner institutions:** None

**Field:** Computer Science and Creative Technologies

**Module type:** Module

**Pre-requisites:** Principles of Programming 2022-23

**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** Develop the necessary transferable skills – e.g. communication, delegation, openness, decision making, flexibility and tolerance

**MO5** Discuss the need for security in the context of system development

**MO6** Reflect emerging software development tools, technologies and methods, e.g. cloud based development and devops.

**Hours to be allocated:** 300

**Contact hours:**

Independent study/self-guided study = 228 hours

Face-to-face learning = 72 hours

Total = 300

**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 coursework comprises a group project assignment with three parts. The first two parts will result in group work and to be submitted as a group whereas the third part will be an individual reflective report to be submitted individually.

The first part covers the analysis of a problem, the specification of its associated requirements and the design of a solution. This will be followed by a short in-class viva. The second part covers the implementation and testing of the design created in the first part. Assessment of this component will be based both on portfolio submissions that include analysis, design, coding (software) and testing documents and a brief group report (up to 700 words) on the software development methodology and applied security measures. This will be followed by an in-class software demonstration. The mark for part 1 and 2 is a group mark, adjusted according to department group work guidelines following tutor assessment of individual contributions to the group.

The final part is an individual reflection (up to 500 words) on the individual contributions, benefits and challenges of conducting this as a group process and use of contemporary tools or technologies.

Students will have the opportunity for formative feedback during practical lab/tutorial sessions.

The resit will also be a group project assignment based on a related problem analysis, requirements specification, design, coding and testing and demonstration of supporting software. The assessment of this component will be based on both portfolio submission that includes analysis, design, coding (software) and testing

documents and a recorded video (up to 8 minutes). The video should include software demonstration, a brief explanation of design, software development methodology and applied security measures.

The final part of the portfolio will be an individual reflection report (up to 500 words) on the individual contributions, benefits and challenges of working in a group project and use of contemporary tools or technologies.

### **Assessment tasks:**

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

Description: The first part covers the analysis of a problem, the specification of its associated requirements and the design of a solution. This will be followed by a short in-class viva. The second part covers the implementation and testing of the design created in the first part. Assessment of this component will be based both on portfolio submissions that include analysis, design, coding (software) and testing documents and a brief group report (up to 700 words) on the software development methodology and applied security measures. This will be followed by an in-class software demonstration. The mark for part 1 and 2 is a group mark, adjusted according to department group work guidelines following tutor assessment of individual contributions to the group.

Weighting: 90 %

Final assessment: No

Group work: Yes

Learning outcomes tested: MO1, MO2, MO3, MO4, MO5

#### **Reflective Piece (First Sit)**

Description: Individual reflective report up to 500 words

Weighting: 10 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO6

#### **Portfolio (Resit)**

Description: The group project assignment is based on a related problem analysis, requirements specification, design, coding and testing and demonstration of supporting software. The assessment of this component will be based on both portfolio submission that includes analysis, design, coding (software) and testing documents and a recorded video (up to 8 minutes). The video should include software demonstration, a brief explanation of design, software development methodology and applied security measures.

Weighting: 90 %

Final assessment: No

Group work: Yes

Learning outcomes tested: MO1, MO2, MO3, MO4, MO5

### **Reflective Piece (Resit)**

Description: Individual reflective report up to 500 words

Weighting: 10 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO6

## **Part 5: Contributes towards**

This module contributes towards the following programmes of study:

Computer Science (Artificial Intelligence) [NepalBrit] BSc (Hons) 2022-23

Computer Science [Villa] BSc (Hons) 2022-23

Computer Science [Villa] BSc (Hons) 2022-23

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

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

Computer Science {Foundation}[Oct][FT][GCET][4yrs] BSc (Hons) 2021-22

Computer Science {Foundation}[Sep][FT][Frenchay][4yrs] BSc (Hons) 2021-22

Computer Science {Foundation}[Sep][FT][Frenchay][4yrs] BSc (Hons) 2021-22

Computer Science {Foundation}[Sep][SW][Frenchay][5yrs] BSc (Hons) 2021-22

Computer Science {Foundation}[Sep][SW][Frenchay][5yrs] BSc (Hons) 2021-22

Computer Science (Artificial Intelligence) {Foundation}[Oct][FT][GCET][4yrs] BSc (Hons) 2021-22

Computer Science {Foundation}[Feb][FT][GCET][4yrs] BSc (Hons) 2021-22

Computer Science (Artificial Intelligence) {Foundation}[Feb][FT][GCET][4yrs] BSc (Hons) 2021-22

Computer Science {Foundation}[Feb][PT][GCET][8yrs] BSc (Hons) 2021-22