



Module Specification

Fundamentals of Software Development

Version: 2023-24, v3.0, 10 Jul 2023

Contents

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

Part 1: Information

Module title: Fundamentals of Software Development

Module code: UFCFQM-30-1

Level: Level 4

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: None

Excluded combinations: None

Co-requisites: None

Continuing professional development: No

Professional, statutory or regulatory body requirements: None

Part 2: Description

Overview: This module aims to provide students with a foundational understanding of software development principles and practices, including topics such as programming paradigms, software design principles, algorithms, and data structures, and testing methodologies.

Features: Not applicable

Educational aims: The purpose of this module is to introduce the students to the fundamental concepts of systems development through programming, computational thinking, and data structures.

They will apply their knowledge of software development concepts to real-world problems, using critical thinking and problem-solving skills to develop effective solutions, including software design, coding, testing, and debugging, through a hands-on programming project.

This will foster an understanding of the importance of software quality, including maintainability, scalability, and reliability, and equip students with the skills to develop high-quality software applications.

This module will prepare students for further study in software development, by providing a solid foundation in fundamental concepts and practical skills that will be useful in more advanced future study.

Outline syllabus: Introduction to software development, including the software development life cycle (SDLC) and Agile methodologies.

Create a sound solution by analysing the software requirements of the business.

Programming fundamentals, including programming paradigms, data types, control structures, and functions.

Data structures, such as arrays, linked lists, stacks, queues, and trees.

Programming paradigms and the reasoning behind selecting it for business requirements (e.g., Event Driven, Object Orientation, Procedural)

Testing facilities and tracking to be able to debug created program code to understand and rectify problems within the code (e.g., white box, black box, unit testing)

Usage of industry standards to create robust and efficient code and understanding the necessity of this practice for both solo and group projects

Part 3: Teaching and learning methods

Teaching and learning methods: Introductory lectures are supported by seminars, case studies, and practical workshops. In addition, this module will be supported by interactive forums and learning tools.

Independent learning includes hours engaged with essential reading, assignment and completion. Study time will be organised each week with a series of both essential and further readings.

This module will be based on ensuring that students' practical skills are developed in programming. Every session will incorporate designated practical work to complete to ensure that students understand and implement principles of good practice.

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

MO1 Explain the stages of a software development lifecycle.

MO2 Design, implement, test, and debug software to meet a requirement's specification, using agreed standards and tools.

MO3 Develop moderately complex software solutions and software modifications to specified requirements.

MO4 Justify the methodology and techniques used to complete a project identifying opportunities for improvement.

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/363CA84E-881E-B1F4-1B7D-9861A48A8EA9.html) via the following link <https://rl.talis.com/3/uwe/lists/363CA84E-881E-B1F4-1B7D-9861A48A8EA9.html>

Part 4: Assessment

Assessment strategy: This module comprises two assessments: a practical programming project and a 15 minute technical interview.

Practical Build:

The programming project is an opportunity for students to demonstrate their practical skills in software development. Students will work on a software development project individually, applying their knowledge of programming languages and software design to develop a functional and well-designed software application.

Students will be given a business specification from which they will produce a solution. They will need to design their systems and apply their knowledge of the development lifecycle models to create a sound system.

The task will include practical software design, development, implementing, testing, and debugging. The testing of the program will need to be robust and thorough, using techniques such as white and black box testing. After the testing, students will be required to modify their existing code base and apply new fixes or modules from their requirements and testing. The program will need to be fully documented and conform to industry standards.

Technical Interview:

The 15 minute technical interview is designed to assess the student's knowledge of fundamental concepts in software development, such as programming languages, data structures, algorithms, software design patterns and software development.

During the technical interview, students will be asked a series of questions related to programming languages, data structures, algorithms, and software design patterns, and they will be expected to provide detailed and accurate responses.

In addition to assessing technical understanding, communication and problem-solving skills, the technical interview will also evaluate students' programming abilities by undertaking a review of the project completed in the first assessment. This will require students to justify their code, methodology, and evaluate how the application could be further improved.

The module will be guided by an experienced instructor, who will provide feedback and support to the students throughout the development process.

The resit opportunities follow the same format as the first submission. Students should be aware that the Technical Interview requires student to review their code from the first project, and a non-submission may limit the areas for discussion in the Interview.

Assessment tasks:

Project (First Sit)

Description: Design, Implement and Test a complete software solution to meet business requirements

Weighting: 70 %

Final assessment: No

Group work: No

Learning outcomes tested: MO2, MO3

Presentation (First Sit)

Description: Technical Interview (15 mins)

Weighting: 30 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO4

Project (Resit)

Description: Design, Implement and Test a complete software solution to meet business requirements

Weighting: 70 %

Final assessment: No

Group work: No

Learning outcomes tested: MO2, MO3

Presentation (Resit)

Description: Technical Interview (15 mins)

Weighting: 30 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO4

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Digital and Technology Solutions (Network Engineer) {Apprenticeship-UCW} [UCW]
BSc (Hons) 2023-24

Digital and Technology Solutions (Data Analyst) {Apprenticeship-UCW} [UCW] BSc
(Hons) 2023-24

Digital and Technology Solutions (Software Engineer) {Apprenticeship-UCW} [UCW]
BSc (Hons) 2023-24

Digital and Technology Solutions (Business Analyst) {Apprenticeship-UCW} [UCW] -
Withdrawn BSc (Hons) 2023-24

Digital and Technology Solutions (Cyber Security Analyst) {Apprenticeship-UCW}
[UCW] - Withdrawn BSc (Hons) 2023-24

Digital and Technology Solutions (Data Analyst) {Apprenticeship-UCW} [UCW] -
Withdrawn BSc (Hons) 2023-24

Digital and Technology Solutions (Software Engineer) {Apprenticeship-UCW} [UCW]
- Withdrawn BSc (Hons) 2023-24

Digital and Technology Solutions (Cyber Security Analyst) {Apprenticeship-UCW}
[UCW] BSc (Hons) 2023-24

Digital and Technology Solutions (Software Engineer) {Apprenticeship-GlosColl}
[GlosColl] BSc (Hons) 2023-24