Module Specification

Object Oriented Software Design and Development I

Version: 2023-24, v2.0, 16 Mar 2023

Contents

Part 1: Information

Module title: Object Oriented Software Design and Development I

Module code: UFCFUM-15-2

Level: Level 5

For implementation from: 2023-24

UWE credit rating: 15

ECTS credit rating: 7.5

Faculty: Faculty of Environment & Technology

Department: FET Dept of Computer Sci & Creative Tech

Partner institutions: None

Delivery locations: Not in use for Modules

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: Students will learn the basic concepts of software design, data structures, programming, problem solving, programming logic, and fundamental software design techniques. This will include a review of traditional and contemporary software development methods including agile development. They will develop a holistic view of software engineering practice including gathering requirements, designing a solution, implementing a solution in a programming language, testing the completed application and deploying the solution to end users.

Features: Not applicable

Educational aims: The purpose of this topic is to introduce the students to the fundamental concepts of systems development through programming, computational thinking and data structures. They will analyse models of application development so that they can understand the key processes related to building functioning applications and appreciate the complexity of application development.

Outline syllabus: Using an industry recognised language students will:

Demonstrate an understanding of object-oriented concepts (e.g. classes, objects, inheritance, polymorphism, encapsulation)

Perform object-oriented analysis and design. This will need to incorporate valid object-oriented designs (e.g. use case, user stories)

Object-oriented Program Development, to a defined business' requirement

Software artefacts

Apply good business practice in all areas of the development life cycle

The use of an appropriate object oriented testing facility 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)

Part 3: Teaching and learning methods

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

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

This unit is practically based and designed to ensure that students understand and develop their skills in advanced programming techniques. Students will use the object-oriented facilities within C++ as a vehicle for this.

Scheduled learning will typically include lectures, seminars, supervision, external visits and an interactive forum.

All students are expected to attend a series of tutorials.

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

MO1 Describe the theory and concepts of the object-oriented paradigm

MO2 Understand software design approaches and patterns and can interpret and implement a given design

MO3 Analyse business and technical requirements and select appropriate solutions

MO4 Create analysis artefacts, such as Use Cases and/or User Stories

MO5 Implement, test, and debug software to meet a requirements specification

MO6 Develop moderately complex software solutions and software modifications to specified requirements

MO7 Debug own code and understand structure of programmes in order to identify and resolve issues

MO8 Identify and apply best practices and standards

Hours to be allocated: 150

Contact hours:

Independent study/self-guided study = 114 hours

Face-to-face learning = 36 hours

Total = 150

Reading list: The reading list for this module can be accessed at readinglists.uwe.ac.uk via the following link https://rl.talis.com/3/uwe/lists/191A5CA5-84FA-5EF6-E16C-CCD8C3E9D807.html

Part 4: Assessment

Assessment strategy: This module is assessed by a combination of techniques: an examination (3 hours) and a practical build.

Exam

Students will be required to sit a 3-hour exam that will require knowledge of the following object-oriented techniques, classes, objects, inheritance, polymorphism and encapsulation.

Students will be required to perform object-oriented design using a taught methodology, against a business specification. They will need to employ various designs such as use case and user stories.

Practical Build

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 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. The program must be fully object-oriented to ensure a sound understanding of the benefits that are associated with object-oriented programming. 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.

Opportunities for formative assessment exist for the assessment strategy used. Verbal feedback and written feedback is given to all students providing a personal platform for improvement.

Assessment components:

Examination (First Sit)
Description: Examination (3 hours)
Weighting: 50 %
Final assessment: Yes
Group work: No
Learning outcomes tested: MO1, MO2, MO3, MO4

Portfolio (First Sit)
Description: Design, implement, test and correct a problem specification
Weighting: 50 %
Final assessment: No
Group work: No
Learning outcomes tested: MO5, MO6, MO7, MO8

Examination (Resit)
Description: Examination (3 hours)
Weighting: 50 %
Final assessment: Yes
Group work: No
Learning outcomes tested: MO1, MO2, MO3, MO4

Portfolio (Resit)

Description: Design, implement, test and correct a problem specification

Weighting: 50 %

Final assessment: No

Group work: No

Learning outcomes tested: MO5, MO6, MO7, MO8

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Digital and Technology Solutions (Software Engineer) {Apprenticeship-UCW} [UCW] BSc (Hons) 2022-23

Digital and Technology Solutions (Data Analyst) {Apprenticeship-UCW} [UCW] BSc (Hons) 2022-23

Digital and Technology Solutions (Business Analyst) {Apprenticeship-UCW} [UCW] BSc (Hons) 2022-23

Digital and Technology Solutions (Cyber Security Analyst) {Apprenticeship-UCW} [UCW] BSc (Hons) 2022-23