Module Specification

Object Oriented Software Design and Development

Version: 2023-24, v3.0, 11 Jul 2023

Contents

Part 1: Information

Module title: Object Oriented Software Design and Development

Module code: UFCFME-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: Software Design and Development 2023-24

Excluded combinations: None

Co-requisites: None

Continuing professional development: No

Professional, statutory or regulatory body requirements: None

Part 2: Description

Overview: Object Oriented Software Design and Development provides students with the essential knowledge and skills required to understand core computer science concepts and construct efficient OO-based software systems.

Features: Not applicable

Educational aims: This module will enable you to:

- Demonstrate an understanding of computer science theory and object-oriented concepts.

- Outline the general trends in software development and identify the perceived advantages of various techniques, including object-oriented approaches.

- Understand and analyse algorithms, data structures, and design principles.

- Create object-oriented designs using a recognised format.

- Create efficient object-oriented software to a required specification.

- Test programs to ensure they meet requirements.

Outline syllabus: Throughout this module students will cover topics including;

Fundamental computer science concepts: algorithms, data structures, and design principles.

Big O notation and algorithm complexity analysis.

Object-oriented concepts.

Trends in software development.

Perceived advantages of various techniques, including object-oriented approaches (e.g., modularity, re-use, iterative development, interactivity, greater client involvement in design, identification of objects, classification, abstraction, encapsulation, inheritance, polymorphism).

Creating object-oriented designs using a recognised format, eg UML.

Implementing efficient object-oriented software to a required specification.

Testing strategies, producing a detailed test plan, and supporting documentation.

Part 3: Teaching and learning methods

Teaching and learning methods: Introductory lectures will cover the fundamentals of computer science theory and the technical underpinning of object-oriented programming for the first assessment.

These will be followed by practical delivery through a series of lessons, workshops, and practical tasks in the classroom to develop the tools and techniques required to complete the practical assessment for this module.

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

MO1 Apply core computer science concepts, eg algorithms, data structures, design principles, Big O notation.

MO2 Demonstrate and apply object-oriented programming concepts including coupling, cohesion and encapsulation.

MO3 Evaluate software development techniques, including object-oriented approaches.

MO4 Develop and document efficient, maintainable and testable object-oriented software applications that adhere to coding standards.

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 via the following link https://rl.talis.com/3/uwe/lists/191A5CA5-84FA-5EF6-E16C-CCD8C3E9D807.html

Part 4: Assessment

Assessment strategy: Object-oriented Software Design & Development is assessed through a single practical portfolio:

The practical portfolio will require students to apply their knowledge of computer science concepts and object-oriented techniques to design, develop and test a software solution. The portfolio is usually comprised of several smaller deliverables, emphasising elegant and efficient code design. to meet the module outcomes.

The completed software solution should adhere to industry best practices and include extensive testing.

The resit opportunity for this module compromises of a re-working of the practical portfolio to meet all learning outcomes.

Tutor-led formative feedback will be available throughout the module.

Assessment tasks:

Portfolio (First Sit)
Description: Practical Portfolio
Weighting: 100 %
Final assessment: Yes
Group work: No
Learning outcomes tested: MO1, MO2, MO3, MO4

Portfolio (Resit)
Description: Practical Portfolio

Weighting: 100 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2, MO3, MO4

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Applied Computing[UCW] BSc (Hons) 2022-23