Module Specification

Computing Practice

Version: 2023-24, v2.0, 27 Mar 2023

Contents

Part 1: Information

Module title: Computing Practice

Module code: UFMFSJ-15-1

Level: Level 4

For implementation from: 2023-24

UWE credit rating: 15

ECTS credit rating: 7.5

Faculty: Faculty of Environment & Technology

Department: FET Dept of Engineering Design & Mathematics

Partner institutions: Gloucestershire College

Field: Engineering, Design and Mathematics

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 concentrates on learning basic syntax and use of an object-oriented Programming, stepwise-refinement approach to design and implementation with exposure to classes, methods, attributes of objects.

Features: Not applicable

Educational aims: This module equips students with the theory and practice of basic syntax of C++ to support a structured approach to program development using object-oriented program and control structures - data types - reuse of basic functions for I/O, string and mathematical manipulation.

Outline syllabus: Typical topics covered include:

Rationale for using C++ in Software Development

C++ language features

Memory allocation / deallocation

Object orientation: inheritance and polymorphism

Exception handling

Templates

Operator overloading

Delegate functions

Compiler directives

Unmanaged code

Automatic vs dynamic memory handling

Standard Template Library

Measuring and analysing performance

Memory alignment, bit manipulation, packing, pooling

Part 3: Teaching and learning methods

Teaching and learning methods: This module will typically involve 3 hours contact time per week. The time will be more or less equally divided between lecture sessions, laboratory sessions.

The module will be taught with a very strong emphasis on practical work and the development of understanding by numerous demonstrations and simple, progressive exercises.

The module will concentrate on teaching basic syntax and use of a class, inheritance, polymorphism, memory management, templates, operator overloading, delegate functions, Compiler directives, structured, stepwise-refinement approach to design and implementation of design tasks.

An extended case-study, supported by focussed laboratory based workshops, will allow the students to follow through an example application from design to implementation, and appreciate the relevance of all the component parts of the module syllabus.

Lectures will be used to introduce concepts, syntax and design methods. Laboratory sessions (workshops) will be used to practice and reinforce the students understanding of these. Students will be expected to work for an equivalent amount of their own time independently on the workshop material, and to independently read their reference book.

Scheduled learning includes lectures and workshops.
Independent learning includes hours engaged with essential reading, assignment preparation and completion etc.

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

MO1 Analyse the impact of using various C++ language features on the compilation process.

MO2 Demonstrate an understanding of the run-time behaviour of a C++ application, and the significance of the call-stack.

MO3 Design and implement object orientated applications that make appropriate use of mechanisms such as polymorphism, templates and delegate functions.

MO4 Identify and manage memory requirements within C++, to develop object-oriented applications which avoid issues such as memory leaks, pointer errors and undefined behaviour.

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://uwe.rl.talis.com/modules/ufmfsk-15-1.html

Part 4: Assessment

Assessment strategy: Assessment will be by practical exercises This strategy has been chosen to develop the student’s knowledge of the theory that they require in order to be successful at the coursework, thus reducing the impact of issues such as collusion.

The coursework will be in the form of a portfolio of work with several practical exercises to be completed throughout the module run. There will be multiple opportunities for formative feedback. All associated learning outcomes will be assessed. A significant percentage of the marks will be awarded for the students demonstrating and explaining their work.

The resit assessment will take the same format as the first sit assessment.

Assessment tasks:

Portfolio (First Sit)
Description: The coursework will be in the form of a portfolio of work containing several practical exercises.
Weighting: 100 %
Final assessment: Yes
Group work: No
Learning outcomes tested: MO1, MO2, MO3, MO4

Portfolio (Resit)
Description: The coursework will be in the form of a portfolio of work containing several practical exercises.
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:

Electronic and Computer Engineering [Frenchay] BEng (Hons) 2023-24

Electronic and Computer Engineering [GlosColl] BEng (Hons) 2022-23

Electronic and Computer Engineering {Apprenticeship-GLOSCOLL} [GlosColl] BEng (Hons) 2022-23