



Module Specification

Program Design and Implementation

Version: 2023-24, v2.0, 03 Jan 2023

Contents

| | |
|--|----------|
| Module Specification | 1 |
| Part 1: Information | 2 |
| Part 2: Description | 2 |
| Part 3: Teaching and learning methods | 3 |
| Part 4: Assessment..... | 5 |
| Part 5: Contributes towards | 6 |

Part 1: Information

Module title: Program Design and Implementation

Module code: UFCEXX-30-0

Level: Level 3

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: Not applicable

Features: Not applicable

Educational aims: See Learning Outcomes

Outline syllabus: Basic programming concepts:

Sequence, selection and iteration constructs; Simple data types, structured data

types: arrays and structs; Access scope of local and global variables; The use of functions with parameters; modularisation: coupling and cohesion and structure charts; Arithmetic and logical operations; Handling strings; Use of pointers to access arrays and structs

Data i/o using keyboard, screen and web; Simple file handling; Operations: searching and sorting

Familiarisation with IDE (integrated Development Environment) and command line operation; Testing; Proof of testing

Polyas approach to problem solving; Algorithmic and strategy problems such as river crossing and NIM

Software design and development:

Requirements analysis, functional and non-functional requirements

Design and development techniques; Top-down vs bottom-up; Functional decomposition; Iterative design, prototyping

Implementation; Testing, verification and validation

Part 3: Teaching and learning methods

Teaching and learning methods: This module will use lectures to introduce new concepts and direct the students to texts and web sources, while associated practical laboratory sessions will expose and explore the material in greater depth. Students will be expected to carry out independent study in parallel with the timetabled periods.

The design and production of programs requires practice if it is to be mastered. This module will include the setting and solving of a series of small problems, each introducing a new idea or technique to be mastered. Some of these will be solved in preparation for classes that will discuss the merits or otherwise of certain

approaches. There will be regular feedback on the exercises to facilitate the development of skills as the module progresses. Lectures will be used to introduce new ideas.

Scheduled learning includes lectures, seminars, tutorials, project supervision, demonstration, practical classes and workshops; fieldwork; external visits; work based learning; supervised time in studio/workshop.

Independent learning includes hours engaged with essential reading, case study preparation, assignment preparation and completion etc. These sessions constitute an average time per level as indicated in the table below. Scheduled sessions may vary slightly depending on the module choices you make.

Placement learning: may include a practice placement, other placement, year abroad.

Activity (hrs)

Contact time (72)

Assimilation and development of knowledge (148)

Coursework preparation (80)

Total study time (300)

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

MO1 Demonstrate understanding, and appropriate use, of a variety of notations for the specification of processing rules and algorithms

MO2 Discuss the relative merits of different programming languages and their use in the development of software for different applications and platforms

MO3 Demonstrate understanding of the functional role of common software development tools and the use of libraries in the development of software

MO4 Demonstrate understanding of the structure and syntax of a high level programming language and the use of data structures and syntactic constructs in the implementation of algorithms

MO5 Use appropriate methods to design, implement and test programs to achieve functional and non-functional requirements, derived from a simple requirements specification

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://uwe.rl.talis.com/modules/ufcexx-30-0.html) via the following link <https://uwe.rl.talis.com/modules/ufcexx-30-0.html>

Part 4: Assessment

Assessment strategy: The assessment will be conducted through two assessment points: (1) A portfolio is to be put together from series of workshop exercises with increasing complexity that aims at developing competency and confidence in the use of a programming environment and basic programming skills. (2) A final written examination, testing depth of understanding and evaluative skills

Resit assessments follow the first sit.

Assessment tasks:

Examination (First Sit)

Description: Examination (3 hours)

Weighting: 40 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2, MO4, MO5

Portfolio (First Sit)

Description: Portfolio of programming exercises

Weighting: 60 %

Final assessment: No

Group work: No

Learning outcomes tested: MO3, MO4, MO5

Examination (Resit)

Description: Examination (3 hours)

Weighting: 40 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2, MO4, MO5

Portfolio (Resit)

Description: Portfolio of programming exercises

Weighting: 60 %

Final assessment: No

Group work: No

Learning outcomes tested: MO3, MO4, MO5

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Mechanical Engineering and Technology {Foundation} [GCET] BEng (Hons) 2023-24

Mechanical Engineering and Technology (Vehicle Technology) {Foundation} [GCET]
BEng (Hons) 2023-24

Mechanical Engineering and Technology (Mechatronics) {Foundation} [GCET] BEng
(Hons) 2023-24

Mechanical Engineering and Technology (Manufacturing) {Foundation} [GCET]
BEng (Hons) 2023-24

Electronics and Telecommunication Engineering {Foundation} [GCET] BEng (Hons)
2023-24

Instrumentation and Control Engineering {Foundation} [GCET] BEng (Hons) 2023-24

Automation and Robotics Engineering {Foundation} [GCET] BEng (Hons) 2023-24

Automation and Robotics Engineering {Foundation} [GCET] DipHE 2023-24

Electronics and Telecommunication Engineering {Foundation} [GCET] DipHE 2023-
24

Instrumentation and Control Engineering {Foundation} [GCET] DipHE 2023-24

Mechanical Engineering and Technology (Manufacturing) {Foundation} [GCET]
DipHE 2023-24

Mechanical Engineering and Technology (Mechatronics) {Foundation} [GCET]
DipHE 2023-24

Mechanical Engineering and Technology (Vehicle Technology) {Foundation} [GCET]
DipHE 2023-24

Mechanical Engineering and Technology {Foundation} [GCET] DipHE 2023-24

Instrumentation and Control Engineering {Foundation} [GCET] BEng (Hons) 2022-23