



Module Specification

Programming [TSI]

Version: 2023-24, v2.0, 20 Mar 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: Programming [TSI]

Module code: UFCFBW-18-0

Level: Level 3

For implementation from: 2023-24

UWE credit rating: 18

ECTS credit rating: 9

Faculty: Faculty of Environment & Technology

Department: FET Dept of Computer Sci & Creative Tech

Partner institutions: Transport and Telecommunication Institute

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: Not applicable

Features: Not applicable

Educational aims: The aim of this module is to enable students to develop algorithmic thinking, learn basic concepts and programming terms, and to become

acquainted with high level programming constructs and software development processes.

Outline syllabus: The module covers the following topic areas:

Methodology and paradigms of programming,
Algorithms. Forms of algorithm representation,
Basic programming constructs,
Introduction to high level programming language (C++),
History of C++,
Composition of a high level programming language: identifiers, keywords, constants,
Composition of C++,
Program structure. Input and output,
Basic data types. Variables and expressions,
Introduction to the programming environment,
Operations. Math functions,
Branching. Conditional and unconditional jumps, multiple choice operators,
Loop operators,
Memory addressing and pointers,
Data structures - arrays. Single and multidimensional arrays,
Basic algorithms for working with arrays,
Work with string data,
User defined functions and types,
Working with files,
Standard Template Library (STL),

Part 3: Teaching and learning methods

Teaching and learning methods: Learning and teaching will be provided to students in two forms: lectures, practical classes and labs. During lectures, theoretical aspects of the course will be provided to students by the teaching staff. Lectures will be supported by presentation published and available to the students on e.tsi.lv under the module section. Also, additional materials, like code examples,

text books, publications on the internet, videos etc will be presented in e.tsi.lv.
During labs, each student receives an individual task to perform.

During practical classes detailed discussion about specific algorithms is conducted & algorithms work demonstration is done by the teaching staff.

C++ is studied, as an example of high-level programming language. In addition to learning activities during taught sessions, students are expected to spend time outside of class on independent learning activities. These might include completing assignment tasks, independent reading, practising new skills on personal projects and watching informative videos, completing self-assessment test etc.

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

MO1 Apply basic principles of algorithmization

MO2 Use basic notations and methods in programming

MO3 Apply basic data structures

MO4 Use high level programming language (as example C++) to complete programming of algorithms

MO5 Understand the development stages of a program

MO6 Decompose programming tasks into smaller logical parts and creation of algorithms that implement subtasks

MO7 Implement algorithms and data processing methods using high level programming language

MO8 Use of C++ operators

MO9 Use development environments and debuggers for program development and testing

Hours to be allocated: 180

Contact hours:

Independent study/self-guided study = 96 hours

Face-to-face learning = 64 hours

Total = 160

Reading list: The reading list for this module can be accessed at [readinglists.uwe.ac.uk](https://rl.talis.com/3/uwe/lists/565B2E25-6B07-C0C8-483D-7D527B92F01F.html?lang=en-gb&login=1) via the following link <https://rl.talis.com/3/uwe/lists/565B2E25-6B07-C0C8-483D-7D527B92F01F.html?lang=en-gb&login=1>

Part 4: Assessment

Assessment strategy: This module assessment is split into two assessments (Exam, Labs):

The practical assignment should be completed individually (i.e. this is not group work) and represents 50% of your final module mark. The practical assignment has 11 elements.

A final 2-hour examination which will assess the students understanding of taught material that forms part of the learning outcomes but cannot easily be assessed through practical tasks. This component represents 50% of final module mark.

A series of 11 practical tasks (labs), exploring basic principles of programming using C++ programming language, which should be completed by the students. An application and its source code should be provided to the teaching staff.

Assessment components:

Examination (First Sit)

Description: Examination (2 hours)

Weighting: 50 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO2, MO3, MO5

Portfolio (First Sit)

Description: series of 11 practical tasks (labs), exploring basic principles of programming using C++ programming language, which should be completed by the students. An application and its source code should be provided to the teaching staff.

Weighting: 50 %

Final assessment: No

Group work: No

Learning outcomes tested: MO1, MO2, MO3, MO4, MO6, MO7, MO8, MO9

Examination (Resit)

Description: Examination (2 hours)

Weighting: 50 %

Final assessment: No

Group work: No

Learning outcomes tested: MO1, MO2, MO3, MO5

Portfolio (Resit)

Description: series of 11 practical tasks (labs), exploring basic principles of programming using C++ programming language, which should be completed by the students. An application and its source code should be provided to the teaching staff.

Weighting: 50 %

Final assessment: No

Group work: No

Learning outcomes tested: MO1, MO2, MO3, MO4, MO6, MO7, MO8, MO9

Part 5: Contributes towards

This module contributes towards the following programmes of study: