STUDENT AND ACADEMIC SERVICES

MODULE SPECIFICATION

Part 1: Information

Module Title	Programming [TSI]		
Module Code	UFCFBW-18-0	Level	Level 3
For implementation from	2020-21		
UWE Credit Rating	18	ECTS Credit Rating	9
Faculty	Faculty of Environment & Technology	Field	Computer Science and Creative Technologies
Department	FET Dept of Computer Sci & Creative Tech		
Module Type:	Standard		
Pre-requisites	None		
Excluded Combinations	None		
Co-requisites	None		
Module Entry Requirements	None		
PSRB Requirements	None		

Part 2: Description

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),

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 & and 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.

Part 3: Assessment

This module assessment is split into two components (A – Exam, B – Labs):
The practical assignment component 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.

B – 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.

First Sit Components	**Final Assessment**	**Element weighting**	**Description**
Examination - Component A	✓	50 %	Examination (2 hours)
Portfolio - Component B		50 %	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.
Resit Components	**Final Assessment**	**Element weighting**	**Description**
Examination - Component A		50 %	Examination (2 hours)
Portfolio - Component B		50 %	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.

Part 4: Teaching and Learning Methods

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

Module Learning Outcomes	Reference
Apply basic principles of algorithmization	MO1
Use basic notations and methods in programming	MO2
Apply basic data structures	MO3
Use high level programming language (as example C++) to complete programming of algorithms	MO4
Understand the development stages of a program	MO5
Decompose programming tasks into smaller logical parts and creation of algorithms that implement subtasks	MO6
Implement algorithms and data processing methods using high level programming language	MO7
Use of C++ operators	MO8
Use development environments and debuggers for program development and testing	MO9

Contact Hours

Independent Study Hours:	
Independent study/self-guided study	96
Total Independent Study Hours:	96
Scheduled Learning and Teaching Hours:	
Face-to-face learning	64
Total Scheduled Learning and Teaching Hours:	64
Hours to be allocated	180
Allocated Hours	160

Reading List	*The reading list for this module can be accessed via the following link:* https://rl.talis.com/3/uwe/lists/565B2E25-6B07-C0C8-483D-7D527B92F01F.html?lang=en-gb&login=1

Part 5: Contributes Towards

This module contributes towards the following programmes of study:

Computer Science and Software Development [Oct][FT][TSI][4yrs] BSc (Hons) 2020-21

Computer Science and Software Development [Oct][PT][TSI][5yrs] BSc (Hons) 2020-21 BSc (Hons) 2020-21

Computer Science and Software Development [Feb][FT][TSI][4yrs] BSc (Hons) 2020-21

Computer Science and Software Development [Feb][PT][TSI][5yrs] BSc (Hons) 2020-21 BSc (Hons) 2020-21