UWE Bristol University of the West of England

MODULE SPECIFICATION

Part 1: Information			
Module Title	Programming in C++		
Module Code	UFCFGL-30-1	Level	Level 4
For implementation from	2019-20		
UWE Credit Rating	30	ECTS Credit Rating	15
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		

Part 2: Description

Educational Aims: See Learning Outcomes

Outline Syllabus: Introduction to computer programming

Systems programming, differences between languages like Java/Javascript and C/C++

C:
Data types
Iteration (for and while)
Selection (if and switch)
Functions
Structs
Boolean logic and bit fields
Pointers and memory management
Linked lists, stacks, and queues (in C style)

C++:
Introduction to object-oriented programming
Classes, objects, and data-encapsulation
Linked lists, stacks, and queues (in C++ style)
Function objects and Anonymous functions

Generic programming (templates)
Generic linked list
Object-oriented design, introduction to UML
Testing and debugging
Open Source software---examples, licenses, and ethics

Teaching and Learning Methods: Laboratory exercises will allow the student to gain familiarization with the tools and techniques required for the implementation and verification of systems built with C++.

Students will be expected to demonstrate self-direction and originality in their learning which will be facilitated through student directed tutorials.

Scheduled learning: in the form of lectures, tutorials, demonstrations and practical classes will comprise 1/3 of the total study time for this module.

Independent learning: will constitute the remaining study time with an expectation that approximately 92 hours will be spent on self-directed study, a further 80 hours in support of the coursework and 32 hours preparation for the presentation.

Contact time: 72 hours
Assimilation and skill development: 140 hours
Undertaking coursework: 88 hours
Total: 300 hours

Part 3: Assessment

Summative assessment is achieved through the demonstration of an innovative solution to a design problem, which will be a program implementation, design (e.g. UML), and testing, along with submission of a log book, which is between 1500 and 2000 words.

Formative assessment will be provided as oral feedback throughout the laboratory sessions particularly with respect to the design development and the log-book entries.

Final summative assessment will be by oral presentation of the software implemented, reflecting back to the log book.

Students will also be assessed in their effective use of the test and verification tools, the quality of their program design and documentation.

The resit assessment will similarly take the form of a presentation, logbook and demonstration of the final product. However in this situation, the presentation will be a video presentation, and the demonstration of the final product will also be via video.

First Sit Components	**Final Assessment**	**Element weighting**	**Description**
Set Exercise - Component B	✓	50 %	Logbook and demonstration of final product
Presentation - Component A		50 %	Oral Presentation
Resit Components	**Final Assessment**	**Element weighting**	**Description**
Set Exercise - Component B	✓	50 %	Logbook and demonstration of final product
Presentation - Component A		50 %	Video presentation

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**
Understand the foundations of system programming, discuss the difference between managed languages such as Java/Javascript and non-managed languages such as C/C++	MO1
Understand and use the basic programming constructs of C/C++	MO2
Manipulate various C/C++ datatypes, such as arrays, strings, and pointers	MO3
Isolate and fix common errors in C++ programs	MO4
Use memory appropriately, including proper allocation/deallocation procedures	MO5
Apply object-oriented approaches to software problems in C++	MO6
Write small-scale C++ programs using the skills developed during the course	MO7
Develop and use test plans	MO8
Understand and put into practice basic source control management, for example Git.	MO9
Discuss and consider the ethical issues around open source software, considering its advantages and disadvantages.	MO10
Discuss and apply trustworthy and secure software development.	MO11

Contact Hours

Independent Study Hours:	
Independent study/self-guided study	204
Total Independent Study Hours:	204
Scheduled Learning and Teaching Hours:	
Face-to-face learning	96
Total Scheduled Learning and Teaching Hours:	96
Hours to be allocated	300
Allocated Hours	300

Reading List

The reading list for this module can be accessed via the following link:

https://uwe.rl.talis.com/modules/ufcfgl-30-1.html

Part 5: Contributes Towards
This module contributes towards the following programmes of study: Forensic Computing and Security {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19 Forensic Computing and Security {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19