STUDENT AND ACADEMIC SERVICES

MODULE SPECIFICATION

Part 1: Information			
Module Title	C++ Development		
Module Code	UFCFK4-30-2	Level	Level 5
For implementation from	2020-21		
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	Programming in C 2020-21		
Excluded Combinations	None		
Co- requisites	None		
Module Entry requirements	None		

Part 2: Description
Overview: Pre-requisites: students must take UFCFF6-30-1 Programming in C. **Educational Aims:** See Learning Outcomes **Outline Syllabus:** Software development process: Unified Modelling Language diagrams: Use-case, Class and Sequence. Source/Version Control. Software Testing procedures. Test-Driven Design Software Design Principles: Cohesion and Coupling Polymorphism and Inheritance Encapsulation Validation and Verification Basic Software Patterns Rationale for using C++ in Software Development

C++ language features:
Memory allocation / deallocation.
Object orientation
Exception handling.
Templates.
Operator overloading.
Compiler directives.

Software development using C++:
Compilation process (trivial vs non-trivial projects).
Compiler representation of language features.
Runtime behaviour / call-stack behaviour.
Using basic and advanced debugging facilities (dump files, expressions, exception handling,
memory examination and tracing).
Practical considerations – IDEs, libraries and SDKs.

Unmanaged code:
Automatic vs dynamic memory handling.
Standard Template Library.
Measuring and analysing performance.
Memory alignment, bit manipulation, packing, pooling.

Teaching and Learning Methods:
Contact time: 72 hours
Assimilation and development of knowledge: 148 hours
Exam preparation: 20 hours
Coursework preparation: 60 hours
Total study time: 300 hours

Lectures will be used to introduce relevant programming concepts whilst being practically
explored within supervised studio sessions guided by tutorial tasks.

A set number of the tutorial tasks are to be completed to form individual lab logbooks.

Aside from the tutorial tasks, students will be set a small number of more challenging tasks to
implement taught concepts, using supplied designs / code / libraries / SDKs where appropriate. It
is expected that the majority of this work will be carried out independently, outside of taught
sessions, though specific sessions will be organised to provide targeted help with these tasks
prior to hand-in.

Part 3: Assessment

Formative assessment:
The tutorial tasks set for the module will be peer and tutor reviewed regularly in studio/practical sessions. Some of
these tutorial tasks will contribute to a logbook (these tasks will be clearly marked).

Summative assessment:
The logbook represents the first summative assessment for this module and is designed to encourage student
engagement. Completed task must be demonstrated to a tutor before they can be signed-off. The number of tasks
that qualify for the logbook is subject to change but is likely to be between 5 and 10.

A small number of more challenging tasks will be set in the form of a small programming project. These tasks form
the second assessment, and will be set in order of increasing complexity. The project will bring together many
aspects of the module materials and assess many learning objectives.

The reason for this assessment strategy is to align assessed tasks with the topics being taught, and distribute
workload for the module across the year.

A final online examination for the module will assess detailed understanding of language mechanisms that form
part of several learning outcomes but cannot easily be assessed through practical tasks.

STUDENT AND ACADEMIC SERVICES

First Sit Components	Final Assessment	Element weighting	Description
Practical Skills Assessment - Component B		30 %	Practical tutorial exercises (Logbook)
Practical Skills Assessment - Component B		45 %	Small Programming Project
Examination (Online) - Component A	✓	25 %	Online Examination (24 hour time-window)
Resit Components	Final Assessment	Element weighting	Description
Practical Skills Assessment - Component B		75 %	Small Programming Project
Examination (Online) - Component A	✓	25 %	Online Examination (24 hour time-window)

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
Analyse the impact of using various C++ language features on the compilation process for non-trivial software projects and make appropriate choices during development	MO1
Demonstrate an in-depth understanding of the run-time behaviour of a C++ application, and the significance of the call-stack	MO2
Deconstruct a programming problem into a set of functional requirements and create basic object-orientated design solutions using an appropriate diagramming technique	MO3
Implement and test C++ applications based on an object-orientated design solution that make appropriate use of mechanisms such as polymorphism and templates	MO4
Discuss the role and significance of external libraries and Software Development Kits (SDKs), their relationship to C++ and their role in cross-platform development	MO5
Apply understanding of issues surrounding memory management within C++, to develop object oriented applications which avoid issues such as memory leaks, pointer errors and undefined behaviour	MO6
Recognise issues related to efficiency and organisation of memory resources within unmanaged code and apply strategies to reduce their impact on run-time performance	MO7
Evaluate object-oriented solutions using principles of good software design and suggest alternative approaches to improve them (where appropriate)	MO8

Contact Hours	Independent Study Hours:	
	Independent study/self-guided study	228
	Total Independent Study Hours:	228

	Scheduled Learning and Teaching Hours:	
	Face-to-face learning	72
	Total Scheduled Learning and Teaching Hours:	72
	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/ufcfk4-30-2.html	

Part 5: Contributes Towards
This module contributes towards the following programmes of study: Computing [Sep][FT][Frenchay][3yrs] BSc (Hons) 2019-20 Computing [Sep][SW][Frenchay][4yrs] BSc (Hons) 2019-20 Computing {Dual} [Aug][FT][Taylors][3yrs] BSc (Hons) 2019-20 Computing {Dual} [Aug][SW][Taylors][4yrs] BSc (Hons) 2019-20 Computing {Dual} [Mar][FT][Taylors][3yrs] BSc (Hons) 2019-20 Computing {Dual} [Mar][SW][Taylors][4yrs] BSc (Hons) 2019-20 Computing {Foundation} [Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19 Computing {Foundation} [Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19 Digital Media {Foundation}[Sep][SW][Frenchay][5yrs] BSc (Hons) 2018-19 Digital Media {Foundation}[Sep][FT][Frenchay][4yrs] BSc (Hons) 2018-19