MODULE SPECIFICATION

Code: UFEEHG-30-2	**Title: Software Design & C++**	**Version: 2008**
Level: 2	**UWE credit rating: 30**	**ECTS credit rating: 15**
Module Type: Standard		
Owning Faculty: Bristol Institute of Technology	**Field: Computer Science**	
Valid from: 1st September 2008	**Discontinued From: 31st August 2009**	
Pre-requisites:	**UFEEHE-30-1 Programming in C OR UFEE5H-20-1 OR UFEE7A-20-1 Introduction to Software Development in C OR UFEETS-20-1 Programming in C**	
Co-requisites:	**None**	
Excluded combinations:	**None**	

Learning Outcomes

On completion of this module a student will typically be able to:-		*Assessed in component(s):*
A. Show a detailed knowledge and understanding of		
i)	recognition of suitable applications for the object-oriented method	B
ii)	using C++ for embedded and real-time application areas	B
B. Demonstrate subject specific skills with respect to		
i)	programming in C++ using available class libraries	B
ii)	the various development and debugging tool kits of C++ development	A, B
iii)	the nature of the C++ runtime environment	A, B
C. Show cognitive skills with respect to		
i)	analysis and design using object-oriented methods and UML documentation	A, B
ii)	interfacing to C++	A, B
D. Demonstrate key transferable skills in		
i)	communication skills	A, B
ii)	self-management skills	A, B

iii)	IT skills in context	A, B
iv)	problem formulation and decision making	A, B

Syllabus Outline

UML and Case Tools: An Introduction to the Concept and use of Rational Rose Classes, derived classes and objects, members, attributes and operations, Data hiding, private, public, private and protected

Constructors and destructors

Classification hierarchies. Inheritance, Polymorphism, virtual functions and overloading

Input / output using streams

Template and container classes.

Introduction to Microsoft Foundation Class library.

C++ development environments. DevStudio IDE, GNU G++ .

Using C++ for embedded and stand alone applications. Problems with memory allocation, virtual functions and run-time environments.

Interfacing C++ to assembler level programs, dealing with tasking and interrupts.

Teaching and Learning Methods

Individual worksheet and group-oriented practical exercises are central to the students\' experience in order to reinforce and extend the lectures and associated readings. The laboratory work includes design and implementation examples, at basic unit and higher system level.

An extended case-study, supported by focussed tutorials and practicals, will allow the students to follow through an example application from design to implementation, and appreciate the relevance of all the component parts of the module syllabus. Examples of case studies could be: design and development of a lab/room booking distributed application using a compliant windows GUI, developing a small object based scheduler/dispatcher in C++ for downloading onto target hardware.

Indicative Reading List

The following list is offered to provide validation panels/accrediting bodies with an indication of the type and level of information students may be expected to consult. As such, its currency may wane during the life span of the module specification. However, CURRENT advice on readings will be available via other more frequently updated mechanisms.

Pressman R.S. (1992). *Software Engineering: A Practitioners Approach (3rd ed)*, McGraw-Hill

Lee, P.A. and Phillips, C. (1996). *The Apprentice C++ Programmer: A Touch of Class*, Int. Thomson Computer Press

Friedman, F.L. , Koffman, E.B. (1994). *Problem Solving, Abstraction, and Design Using C++*, Addison-Wesley

Capper, D.M. (1994). *C++ for scientists, engineers and mathematicians*, Springer-Verlag

Barr M. (2000). *C and C++ for Embedded Systems*, O'Reilly

Barton, J.J.; Nackman, L.R. (1994). *Scientific and Engineering C++*, Addison-Wesley

Hanly, J.R. (1997). *Essential C++ for Engineers and Scientists*, Addison-Wesley

Assessment

Weighting between components A and B A: 50% B: 50%

ATTEMPT 1
First Assessment Opportunity

Element Description	**% of Component**	**% of Assessment**
Component A (Controlled Conditions)		
Examination	100%	50%
Component B		
Coursework 1.	50%	25%
Coursework 2.	50%	25%

ATTEMPT 2
Second Assessment Opportunity Attendance at taught classes <u>is not</u> required.

Element Description	**% of Component**	**% of Assessment**
Component A (Controlled Conditions)		
Examination	100%	50%
Component B		
Coursework	100%	50%