UNIVERSITY OF KENT AT CANTERBURY

MODULE SPECIFICATION

Title: CO889 C++ Programming

Department responsible: School of Computing

Start date: September 2010

Cohort of students (onwards) to which module will be applicable: 2010-11

Number of students: 30

Modules to be withdrawn: None

Level of the module: M

Number of credits: 15 (7.5 ECTS)

Period taught: Autumn Term

Prerequisite and co-requisite modules

Prerequisite: CO871 Advanced Java for Programmers

The programmes of study to which the module contributes

Portfolio of Taught Postgraduate Programmes in Computing (compulsory for MSc Advanced Programming for Multi-core Systems and MSc Advanced Software Development)

Subject-specific learning outcomes

On successful completion of this module, students will:

a) have experience with, and an understanding of, programming in a low-level object-oriented language to an advanced level; (A1, A2, A4, A16, C11);

b) be able to design and implement object-oriented solutions to a wide range of programming problems, using standard and 3rd-party libraries where appropriate; (A1, A2, A16, C1, C2, C3, C11);

c) have an understanding of the differences between object-oriented languages and the features they provide; (A3);

d) have experience of using standard tools to diagnose errors in object-oriented applications; (C1, C3, C11);

e) be able to understand and modify existing complex object-oriented applications; (A1, C2, C3, C11).

Generic learning outcomes

On successful completion of this module, students will be able to:

f) manage their own time effectively, in the completion of coursework and private study; (D1, D2, D3);

g) identify appropriate solutions to technical problems and be able to apply these; (B1, B2, B3, B4);

h) communicate technical issues with specialist and non-specialist audiences; (D6).
Synopsis of the curriculum

- Introduction to the C and C++ languages, relating to their previous knowledge of Java, including interfaces, classes, abstract classes, inheritance, interfaces, overloading and templates;
- Review of command-line based development, including compiling, debugging and makefiles;
- C++ specific items, including multiple inheritance, namespaces, friend classes, virtual methods, pointers, casting operators and explicit memory management;
- Low-level programming in C++, covering considerations for inline assembly, linkage with other languages and mobile platform development;
- High-level programming in C++, covering large scale applications, efficient programming (e.g. games) and design patterns;
- Coverage of standard libraries, where to look for these, and how to use them;

Indicative Reading List:


Learning and Teaching Methods

All learning outcomes will be achieved through a combination of lectures, supervised practical classes and private study, with further assistance provided electronically via discussion forums and the web.

This module represents a total of 150 study hours, including a total of 30 contact hours for lectures and practical classes.

Assessment methods

Students are graded on a percentage scale with 40% as the pass mark.

All learning outcomes are assessed by coursework and an unseen written examination. The weightings are as follows:

- 50% written examination, assessing outcomes (a,c,d,f,g,h)
- 50% coursework, assessing outcomes (a,b,c,d,e,f,g,h)

Supervised laboratory classes will provide support for students during work on assessments. The coursework will consist of a number of distinct pieces of work, submitted for assessment throughout the term. Coursework will consist of both individual and group programming.

Implications for learning resources, including staff, library, IT and space:

Teaching will be undertaken by existing academic staff from the School of Computing using timetabled university lecture/seminar rooms and existing teaching space within the School.
Students will require normal access to university library and IT resources. A small number of core texts may need to be purchased. Additional software may also be needed but this is free for teaching purposes and will run on existing PCs.

A statement confirming that, as far as can be reasonably anticipated, the curriculum, learning and teaching methods and forms of assessment do not present any non-justifiable disadvantage to students with disabilities.

The School of Computing recognises and has embedded the expectations of SENDA, and supports students with a declared disability or special (educational) need in its teaching, through the establishment of Inclusive Learning Plans agreed between student, department and the Disability Support Unit. We will liaise with the Disability Support Unit in order to provide specialist support where needed.

Statement by the Director of Learning and Teaching: "I confirm I have been consulted on the above module proposal and have given advice on the correct procedures and required content of module proposals"

..........................................................................................................................................
Director of Learning and Teaching Date

Statement by the Head of Department: "I confirm that the Department has approved the introduction of the module and will be responsible for its resourcing"

....................................................................................................................................
Head of Department Date

Last updated: 22 March 2010