

MODULE SPECIFICATION

1. Title of module:

CO541 Object Oriented Programming

Department responsible for management:

Computing

Start date:

September 2006

The cohort of students (onwards) to which the module will be applicable.

2009/10

Number of students:

30

Modules to be withdrawn:

None

Level:

I

Number of credits:

30

Terms during which module taught:

1

Prerequisites and co-requisites:

Prerequisite: CO331 Visual Programming

Programmes of study:

BSc Business Information Technology
BSc Information Technology
and their Year in Industry counterparts.

Subject-specific learning outcomes and relationship to programme learning outcomes:

Students who successfully complete this module will be able to:

- a. Use an object-oriented programming language to write programs. [A2]
- b. Write and test programs with the support of an integrated development environment. [A2, A4, C2]
- c. Choose appropriate data structures for various programming purposes [A2, A4]

- d. Use object-oriented analysis, design and implementation to identify and solve practical programming problems. [A4, C2]
- e. Discuss the quality of solutions through consideration of issues such as encapsulation, cohesion and coupling. [C2]
- f. Use effectively a range of software development tools, such as an integrated development environment, text editor and compiler. [C4]
- g. Use advanced features of an object-oriented programming language, such as inheritance and graphical libraries, to write programs. [A2]
- h. Use object-oriented analysis, design and implementation with a minimum of guidance, to recognise and solve practical programming problems involving inheritance hierarchies. [A4, C2]
- i. Design appropriate interfaces between modular components. [B5]
- j. Evaluate the quality of competing solutions to programming problems. [A4, C2]
- k. Evaluate possible trade-offs between alternative solutions, for instance those involving time and space differences. [C2]

Generic learning outcomes and relationship to programme learning outcomes:

- Students who successfully complete this module will be able to:
- Demonstrate comprehension of the trade-offs involved in design-choices. [B1]
- Make effective use of IT facilities for solving problems. [D3]
- Be able to manage their own learning and development, through self-directed study and working on continuous assessment. [D5]
- Make appropriate choices when faced with trade-offs in alternative designs. [B1]
- Recognise and be guided by social, professional and ethical issues and guidelines and the general contexts in which they apply. [B6]
- Deploy appropriate theory and practices in their use of methods and tools. [B5]

Synopsis of curriculum

This module builds on students' previous knowledge of basic programming to provide a comprehensive approach to object-oriented software development. Fundamentals of classes and objects are key features of class descriptions: constructors, methods and fields are reviewed. Method implementation through assignment, selection control structures, iterative control structures and other statements are introduced. Collection objects are also covered and the availability of library classes as building blocks.

To provide a deeper understanding of and facility with object-oriented program design and implementation, more advanced features of object-orientation, such as inheritance, abstract classes, exceptions, input-output are also covered. These allow an application-level view of design and implementation to be explored.

Throughout the course, the quality of class and application design and the need for a professional approach to software development is emphasized. Testing and debugging feature are one aspect of this.

Indicative reading list:

- "Objects first with Java - A practical introduction using BlueJ" second edition, David J. Barnes and Michael Kölling, Pearson Education, 2009.
- "The Java Programming Language", Ken Arnold, James Gosling and David Holmes, Addison-Wesley, 2000.
- "Effective Java: Programming Language Guide", Joshua Bloch, Addison-Wesley, 2001.

Learning and teaching methods.

Acquisition is through lectures supported by supervised classes. Self-directed learning is facilitated by directed reading and web-based material. Two one-hour lectures and three one-hour classes per week and one one-hour seminar per week. 300 total study hours spent acquiring practical facility understanding of with object-oriented design and programming.

Assessment methods.

Assessment is through a combination of unseen written examination (50%) and assessed coursework (50%).

The coursework assesses the ability to use software development tools and IT facilities and to write and test solutions to practical programming problems. Examinations assess the ability to read, understand, modify and write programs. Both examinations and coursework assess the ability to use advanced programming features to write good quality solutions to practical programming problems. The assessment criteria provide for evaluation of the appropriateness of design choices, and the application of relevant theory and practice. Together coursework and examination assess all the generic and specific learning outcomes.

Implications for learning resources:

None.

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.

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.

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

August 2009