

---

## CO2005 Object-Oriented Programming Using C++

**Credits:** 10    **Convenor:** Dr. S. Yang    **Semester:** 2<sup>nd</sup>

---

<b>Prerequisites:</b>	<i>Essential:</i> CO1003, CO1004, CO1011	<i>Desirable:</i> CO2006
<b>Assessment:</b>	<i>Coursework:</i> 40%	<i>Two hour exam in May/June:</i> 60%
<b>Lectures:</b>	15 hours	<b>Problem Classes:</b> 5 hours
<b>Laboratories:</b>	20 hours	<b>Private Study:</b> 35 hours

---

### Subject Knowledge

**Aims** This module teaches the basic principles of object-oriented programming, design and testing.

**Learning Outcomes** Students should be able to: demonstrate understanding of the basic components of an object-oriented program including methods and attributes, the distinction between classes and instances, the structures required to write basic algorithms, the components of simple text and graphics based interfaces. They should be able to apply the design process using basic object-oriented design notation, and be able to explain the applicability and effectiveness of various basic software testing techniques.

**Methods** Class sessions, recommended textbook, worksheets, automated feedback and extensive web support.

**Assessment** Marked coursework, written examination.

### Skills

**Aims** Develop problem solving skills by developing object-oriented programs in C++.

**Learning Outcomes** Students will be able to: breakdown simple programming goals into object-oriented components, propose and evaluate different designs for solving problems using knowledge of fundamental programming techniques and the facilities available in the C++, and according to the relative space and time efficiency of the proposed solutions, implement the solution using C++, and test and evaluate the finished code.

**Methods** Class sessions together with worksheets.

**Assessment** Marked coursework and traditional written examination.

---

**Explanation of Prerequisites** It is essential that students should have a thorough grasp of imperative programming before attempting this module. It is assumed that all students have a good working knowledge of Java, up to and including the use of inheritance, interfaces and exceptions.

It is beneficial if students taking this module have an understanding of the object-oriented design philosophy, and are familiar with the fundamental concepts of the object-oriented paradigm.

**Course Description** Over the past 15 years C++ has become one of the world's most popular programming languages, due to its potential for producing efficient and compact code taking direct advantage of the powerful *object-oriented* philosophy. As such today's computer scientists and software engineers should be familiar with the use of its central features. This module is intended to give the student a basic grasp of its use for object-oriented programming.

**Detailed Syllabus** Overview of C++ and comparison with Java, highlighting the differences between the type systems, functions, parameter passing, memory management issues; pointer and reference; constructing C++ class, operator overloading in C++ class; inheritance, dynamic and static method invocations; exceptions, input and output.

### Reading List

- [A] Mark Allen Weiss, *C++ for Java Programmers, International Edition*; ISBN: 0131911635, Pearson Prentice Hall 2004.
- [C] Timothy Budd, *C++ for Java Programmers*; ISBN: 0201612461, Addison Wesley Longman 1999.
- [C] Timothy Budd, *Data Structures in C++, using the Standard Template Library*; ISBN: 0201308797, Addison Wesley, 1998.
- [C] Bjarne Stroustrup, *The C++ Programming Language, 3rd edition*; ISBN: 0201889544, Addison-Wesley, 1997.

**Resources** Course notes, web page, study guide, worksheets, lecture rooms with computer projection facilities and OHPs, past examination papers.

**Module Evaluation** Course questionnaires, course review.