



MONASH University

**FIT3126**  
**Applications with C++**

**Unit Guide**

**Semester 2, 2009**

The information contained in this unit guide is correct at time of publication. The University has the right to change any of the elements contained in this document at any time.

*Last updated : 22 Jul 2009*

# Table of Contents

<u>FIT3126 Applications with C++ - Semester 2, 2009</u> .....	1
<u>Chief Examiner</u> .....	1
<u>Lecturer(s) / Leader(s)</u> .....	1
<u>Caulfield</u> .....	1
<u>Introduction</u> .....	2
<u>Unit synopsis</u> .....	2
<u>Learning outcomes</u> .....	2
<u>Contact hours</u> .....	2
<u>Unit relationships</u> .....	2
<u>Prerequisites</u> .....	2
<u>Prohibitions</u> .....	3
<u>Relationships</u> .....	3
<u>Teaching and learning method</u> .....	4
<u>Timetable information</u> .....	4
<u>Tutorial allocation</u> .....	4
<u>Unit Schedule</u> .....	4
<u>Unit Resources</u> .....	5
<u>Prescribed text(s) and readings</u> .....	5
<u>Recommended text(s) and readings</u> .....	5
<u>Required software and/or hardware</u> .....	5
<u>Equipment and consumables required or provided</u> .....	5
<u>Study resources</u> .....	5
<u>Assessment</u> .....	7
<u>Overview</u> .....	7
<u>Faculty assessment policy</u> .....	7
<u>Assignment tasks</u> .....	7
<u>Examination</u> .....	8
<u>Due dates and extensions</u> .....	8
<u>Late assignment</u> .....	8
<u>Return dates</u> .....	8
<u>Appendix</u> .....	9

# **FIT3126 Applications with C++ - Semester 2, 2009**

## **Chief Examiner:**

**Dr Phu Le**

Fax: +61 3 9903 1247

## **Lecturer(s) / Leader(s):**

**Caulfield**

**Mr Jay Zeal**

## Introduction

Welcome to FIT3126 Applications with C++. This is a 6 point unit.

## Unit synopsis

The main topics covered in this unit are: Object-oriented design and programming in C++ which include object identification, data encapsulation, inheritance, and polymorphism; concurrent programming techniques which include principles of concurrency, safety and liveness design, multithreaded programming in C++; network and distributed software construction principles and techniques which include network software performance, client-server software construction, socket and RPC programming in C++; and study and compare C++ with Java and Ada in terms of large software constructions.

## Learning outcomes

By the completion of this unit students will have a theoretical and conceptual understanding of:

1. object-oriented design and programming with C++ for large software development;
2. concurrent design and programming with programming languages which do not have built-in features to support concurrency like C++;
3. how to develop high performance software with programming languages that do not support concurrency such as C++;
4. and good experience of network and distributed programming using C++.

At the completion of this unit, students will:

1. Appreciate OO application design and implementation with C++;
2. Appreciate design and implementation techniques of concurrent applications with C++;
3. Appreciate design and implementation techniques of network applications with C++;
4. have the practical skill of object identification in software development;
5. have the practical skill of concurrent design with mutual exclusion, deadlock free software construction, live lock avoidance, and efficient task communications;
6. achieved good programming skill with multithreading, semaphores, thread communication and synchronisation mechanism;
7. Achieved good experience of network and distributed programming using OO programming, sockets and C++;
8. Achieved real experience to develop high performance applications with C++ using OO programming together with concurrency and networking.

## Contact hours

4 x contact hrs/week

## Unit relationships

### Prerequisites

CPE1001 or CSE1202 or FIT1002 or equivalent

## **Prohibitions**

(Translation set CSE3400), CSE4530, FIT4033

## **Relationships**

FIT3126 is an elective unit.

Before attempting this unit you must have satisfactorily completed

CPE1001 or CSE1202 or FIT1002 or equivalent.

Students must have the following background:

- Fundamental programming
- Introduction to OO design and programming
- Operating system and networking

You may not study this unit and CSE3400 or CSE4530 in your degree.

## Teaching and learning method

2 hour lecture per week

2 hour tutorial per week

on-line information; discussion group; consultation

## Timetable information

For information on timetabling for on-campus classes please refer to MUTTS, <http://mutts.monash.edu.au/MUTTS/>

## Tutorial allocation

On-campus students should register for tutorials/laboratories using the Allocate+ system:

<http://allocate.cc.monash.edu.au/>

## Unit Schedule

Week	Topic	Key dates
1	Introduction to Applications Programming - Basic Concepts	
2	Introduction to C++	
3	Object Oriented Programming Design and Construction	
4	Object Oriented Programming continued	
5	Introduction to Concurrent Programming	
6	Concurrent Programming Design Issues	
7	Concurrent Programming continued	Assignment 1 due
8	Introduction to Network and Distributed Programming	
9	Network and Distributed Programming with C++	
10	Network and Distributed Programming with C++ continued	
Mid semester break		
11	C++ Strengths and Weaknesses for OO, DP and CP	
12	Reading in Applications Programming	Assignment 2 due
13	revision	

## Unit Resources

### Prescribed text(s) and readings

- Stroustrup, B. The C++ Programming Language (3rd ed.), Addison-Wesley, 1997
- Easttom, C++ Programming Fundamentals, Charles River Media, 2003.
- Booch G., Object Oriented Analysis and Design, Benjamin Cummings, Latest Ed..
- Atkinson C., Object-Oriented Reuse, Concurrency and Distribution, Addison-Wesley, Latest Ed..
- Ben-Ari M., Principles of Concurrent and Distributed Programming, Prentice Hall, Latest Ed..
- Hughes C., Parallel and Distributed Programming with C++, Addison-Wesley, Latest Ed..
- Gregory A., Concurrent Programming: Principles and Practice, Benjamin Cummings, Latest Ed..
- Kleiman S., Shah D. and Smaalders B., Programming with Threads. Prentice Hall, Latest Ed..
- Lea D., Concurrent Programming in Java: Design Principles and Patterns, Addison-Wesley. Latest Ed..

Text books are available from the Monash University Book Shops. Availability from other suppliers cannot be assured. The Bookshop orders texts in specifically for this unit. You are advised to purchase your text book early.

### Recommended text(s) and readings

- Easttom, C++ Programming Fundamentals, Charles River Media, 2003.
- Booch G., Object Oriented Analysis and Design, Benjamin Cummings, Latest Ed..
- Atkinson C., Object-Oriented Reuse, Concurrency and Distribution, Addison-Wesley, Latest Ed..
- Ben-Ari M., Principles of Concurrent and Distributed Programming, Prentice Hall, Latest Ed..
- Hughes C., Parallel and Distributed Programming with C++, Addison-Wesley, Latest Ed..
- Gregory A., Concurrent Programming: Principles and Practice, Benjamin Cummings, Latest Ed..
- Kleiman S., Shah D. and Smaalders B., Programming with Threads. Prentice Hall, Latest Ed..
- Lea D., Concurrent Programming in Java: Design Principles and Patterns, Addison-Wesley. Latest Ed..

### Required software and/or hardware

GNU gcc compiler set: g++ (compiler) and gdb (debugger)

Make build management tool.

Tread and Socket libraries (supplied).

### Equipment and consumables required or provided

Students studying off-campus are required to have the minimum system configuration specified by the Faculty as a condition of accepting admission, and regular Internet access. On-campus students, and those studying at supported study locations may use the facilities available in the computing labs. Information about computer use for students is available from the ITS Student Resource Guide in the Monash University Handbook. You will need to allocate up to **n** hours per week for use of a computer, including time for newsgroups/discussion groups.

### Study resources

Study resources we will provide for your study are:

Weekly detailed lecture notes outlining the learning objectives, discussion of the content, required readings and exercises;

FIT3126 Applications with C++ - Semester 2, 2009

Weekly tutorial exercises;

A sample exam;

- Discussion groups;
- This Unit Guide outlining the administrative information for the unit;
- The unit web site on MUSO, where resources outlined above will be made available.



## Assessment

### Overview

Examination: 50%  
Practical work : 50%

### Faculty assessment policy

To pass a unit which includes an examination as part of the assessment a student must obtain:

- 40% or more in the unit's examination, and
- 40% or more in the unit's total non-examination assessment, and
- an overall unit mark of 50% or more.

If a student does not achieve 40% or more in the unit examination or the unit non-examination total assessment, and the total mark for the unit is greater than 44% then a mark of no greater than 44-N will be recorded for the unit.

### Assignment tasks

#### Assignment coversheets

Assignment coversheets are available via "Student Forms" on the Faculty website:

<http://www.infotech.monash.edu.au/resources/student/forms/>

You MUST submit a completed coversheet with all assignments, ensuring that the plagiarism declaration section is signed.

**Assignment submission and return procedures, and assessment criteria will be specified with each assignment.**

#### • Assignment task 1

**Title:**

Assignment 1

**Description:**

Object Oriented Programming and Concurrency

**Weighting:**

20%

**Due date:**

Week 7

#### • Assignment task 2

**Title:**

Assignment 2

**Description:**

Object Oriented Programming and Network Programming

**Weighting:**

20%

**Due date:**

Week 12

### • **Assignment task 3**

**Title:**

Lab work

**Description:**

Weekly lab tasks to be completed during 2 hour lab session

**Weighting:**

10%

**Due date:**

Must be completed during weekly lab session

## **Examination**

- **Weighting:** 50%

**Length:** 3 hours

**Type (open/closed book):** Closed book

**See Appendix for End of semester special consideration / deferred exams process.**

## **Due dates and extensions**

Please make every effort to submit work by the due dates. It is your responsibility to structure your study program around assignment deadlines, family, work and other commitments. Factors such as normal work pressures, vacations, etc. are not regarded as appropriate reasons for granting extensions. Students are advised to NOT assume that granting of an extension is a matter of course.

Students requesting an extension for any assessment during semester (eg. Assignments, tests or presentations) are required to submit a Special Consideration application form (in-semester exam/assessment task), along with original copies of supporting documentation, directly to their lecturer within two working days before the assessment submission deadline. Lecturers will provide specific outcomes directly to students via email within 2 working days. The lecturer reserves the right to refuse late applications.

A copy of the email or other written communication of an extension must be attached to the assignment submission.

Refer to the Faculty Special consideration webpage or further details and to access application forms:  
<http://www.infotech.monash.edu.au/resources/student/equity/special-consideration.html>

## **Late assignment**

Assignments received after the due date will be subject to a penalty of 5 marks per day, including weekends. Assignments received later than one week (seven days) after the due date will not normally be accepted. In some cases, this period may be shorter if there is a need to release sample solutions.

This policy is strict because comments or guidance will be given on assignments as they are returned, and sample solutions may also be published and distributed, after assignment marking or with the returned assignment.

## **Return dates**

Students can expect assignments to be returned within two weeks of the submission date or after receipt, whichever is later.

## Appendix

Please visit the following URL: <http://www.infotech.monash.edu.au/units/appendix.html> for further information about:

- Continuous improvement
- Unit evaluations
- Communication, participation and feedback
- Library access
- Monash University Studies Online (MUSO)
- Plagiarism, cheating and collusion
- Register of counselling about plagiarism
- Non-discriminatory language
- Students with disability
- End of semester special consideration / deferred exams