FIT2071
Foundations of C++

Unit Guide

Semester 1, 2015

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: 21 Jan 2015

Table of Contents

FIT2071 Foundations of C++ - Semester 1, 2015

Following on from FIT1040, this unit introduces the C++ language to students. The unit provides a foundational understanding of the C++ language, regarding syntax and applied practice, with a focus on object-oriented design principles. Fundamental programming control structures in the C++ context will be presented, as will object oriented design as a process for program design and problem solving. More advanced object-oriented programming topics such as inheritance and polymorphism will also be covered. Other C++ fundamentals such as pointers and the STL will be presented, as will implementations of the main algorithms and data structures used in problem solving.

Mode of Delivery

Caulfield (Day)

Workload Requirements

Minimum total expected workload equals 12 hours per week comprising:

(a.) Contact hours for on-campus students:

- Two hours of lectures
- One 2-hour tutorial

(b.) Additional requirements (all students):

- A minimum of 8 hours independent study per week for completing lab and project work, private study and revision.

See also Unit timetable information

Unit Relationships

Prerequisites

FIT1040 or FIT1002

Chief Examiner

Dr Matthew Butler

Campus Lecturer

Caulfield

Matthew Butler

Tutors

Caulfield

Matthew Butler

Yohann Pitrey

Kevin Leo

Jason Haasz

Your feedback to Us

Monash is committed to excellence in education and regularly seeks feedback from students, employers and staff. One of the key formal ways students have to provide feedback is through the Student Evaluation of Teaching and Units (SETU) survey. The University's student evaluation policy requires that every unit is evaluated each year. Students are strongly encouraged to complete the surveys. The feedback is anonymous and provides the Faculty with evidence of aspects that students are satisfied and areas for improvement.

For more information on Monash's educational strategy, see:

www.monash.edu.au/about/monash-directions/ and on student evaluations, see: www.policy.monash.edu/policy-bank/academic/education/quality/student-evaluation-policy.html

Previous Student Evaluations of this Unit

Previous feedback for the unit has been very positive, so many of the aspects have been left the same. In the past students have particualrly liked the assignment structure as well as a focus on more fun game-like activities. New assignments have been created in this format, to encourage further engagement with the content.

Students also responded positively to the lectures being primarily focused on working through code and examples, rather than simply running through dot points from slides. This will continue this year, with a greater focus placed on real-time coding and interactivity with students rather than running through slides.

The use of clickers was introduced in 2014 in an informal manner to increase engagement. Many students responded positively to their use as another way to discuss the concepts. Many students however did not purchase clickers in 2014 as there were no marks allocated to participation. As students should have clickers from undertaking previous units, the use of clickers will continue, still in an informal way without marks attached.

If you wish to view how previous students rated this unit, please go to https://emuapps.monash.edu.au/unitevaluations/index.jsp

Academic Overview

Learning Outcomes

At the completion of this unit students will be able to:

- construct and compile working programs using C++ syntax fundamentals;
- apply object-oriented design principles, including inheritance and polymorphism, to solve programming problems;
- create C++ programs using pointers to demonstrate an understanding of efficient memory use and management;
- troubleshoot C++ program code using an Integrated Development Environment and its tools.

Unit Schedule

Week	Activities	Assessment
0		No formal assessment or activities are undertaken in week 0
1	An Introduction to Syntactic Programming: From Scribble to C++, IDEs	
2	Syntax Building Blocks 1: Data Types, I/O, Sequence	
3	Syntax Building Blocks 2: Selection, Repetition, Arrays	
4	C++ Fundamentals: OO, Program Structure, Functions, STL	Lab Portfolio Submission 1: due Tuesday 24 March 2015
5	OO Design Fundamentals: Classes, Data Members, Member Functions, Constructors	
6	Data Storage: 2D Arrays, Vectors, Linked Lists	
7	Pointers and Memory Management	
8	Advanced OO Design 1: Inheritance, Overloading, Design Patterns	Lab Portfolio Submission 2: due Tuesday 28 April 2015
9	Advanced OO Design 2: Polymorphism, Virtual Functions, Abstract Classes	
10	OO Design in Practice	
11	Further Data Structures and Algorithms	
12	Case Study and Revision	Major Assignment: due Tuesday 26 May 2015
	SWOT VAC	No formal assessment is undertaken in SWOT VAC
	Examination period	LINK to Assessment Policy: http://policy.monash.edu.au/policy-bank/academic/education/assessment/assessment-in-coursework-policy.html

*Unit Schedule details will be maintained and communicated to you via your learning system.

Teaching Approach

Lecture and tutorials or problem classes

This teaching and learning approach provides facilitated learning, practical exploration and peer learning.

In lecture engagement will be enhanced through the use of "clicker" technology to help explore concepts and practical application in a much more interactive manner.

Assessment Summary

Examination (3 hours): 60%; In-semester assessment: 40%

Assessment Task	Value	Due Date

Unit Schedule

Lab Portfolio Submission 1	10%	Tuesday 24 March 2015
Lab Portfolio Submission 2	10%	Tuesday 28 April 2015
Major Programming Assignment	20%	Tuesday 26 May 2015
Examination 1	60%	To be advised

Assessment Requirements

Assessment Policy

Faculty Policy - Unit Assessment Hurdles
(http://intranet.monash.edu.au/infotech/resources/staff/edgov/policies/assessment-examinations/assessment-hurd

Academic Integrity - Please see resources and tutorials at
http://www.monash.edu/library/skills/resources/tutorials/academic-integrity/

Assessment Tasks

Participation

- **Assessment task 1**

Title:
Lab Portfolio Submission 1

Description:
This task comprises the first submission of your laboratory portfolio. Focus will be on demonstrating understanding of basic C++ syntax to ensure that you are successfully making the transition to using syntactic programming in problem solving.

Students will be required to compile a portfolio of major laboratory tasks for submission for assessment. Details of each portfolio component will be clearly indicated in the laboratory tasks for each week. These components will then be combined to form one complete program for assessment.

This submission contains the weekly folio tasks for Weeks 1 to 3 combined together.

Weighting:
10%

Criteria for assessment:
Explicit assessment criteria will be provided in the assignment brief, however students will be assessed on the following broad criteria:

- ♦ Meeting functional requirements as described in the assignment description
- ♦ Demonstrating a solid understanding of C++ concepts, including good practice
- ♦ Demonstrating an understanding of specific C++ concepts relating to the assignment tasks, including control structures and the standard template library
- ♦ Following the unit Programming Style Guide
- ♦ Creating solutions that are as efficient and extensible as possible

Feedback will be provided on your progress to facilitate for improvements in the second set of portfolio pieces.

Late assignments will incur a 5% penalty per late day (including weekends), and may be submitted up to a maximum of 7 days late. After this time submissions will not be accepted without prior arrangement with the unit leader.

Note that submitted code MUST compile, otherwise ZERO marks will be awarded.

Due date:
Tuesday 24 March 2015

- **Assessment task 2**

Title:

Lab Portfolio Submission 2

Description:

This task comprises the second submission of your laboratory portfolio. Focus will be on demonstrating understanding of more complex C++ fundamentals, such as OO design and the use of pointers.

Students will be required to compile a portfolio of major laboratory tasks for submission for assessment. Details of each portfolio component will be clearly indicated in the laboratory tasks for each week. These components will then be combined to form one complete program for assessment.

This submission contains the weekly folio tasks for Weeks 4 to 7 combined together.

Weighting:

10%

Criteria for assessment:

Explicit assessment criteria will be provided in the assignment brief, however students will be assessed on the following broad criteria:

- ♦Meeting functional requirements as described in the assignment description
- ♦Demonstrating a solid understanding of C++ concepts, including good practice
- ♦Demonstrating an understanding of specific C++ concepts relating to the assignment tasks, including advanced object design and implementation, and pointers and memory management
- ♦Following the unit Programming Style Guide
- ♦Creating solutions that are as efficient and extensible as possible

Late assignments will incur a 5% penalty per late day (including weekends), and may be submitted up to a maximum of 7 days late. After this time submissions will not be accepted without prior arrangement with the unit leader.

Note that submitted code MUST compile, otherwise ZERO marks will be awarded.

Due date:

Tuesday 28 April 2015

- **Assessment task 3**

Title:

Major Programming Assignment

Description:

Students will be required to complete a major programming assignment, encompassing all concepts covered in the unit. Focus will be on developing a large, multi-class program. The full assignment brief will be placed on the unit website.

Weighting:

20%

Criteria for assessment:

Explicit assessment criteria will be provided in the assignment brief, however students will be assessed on the following broad criteria:

- ♦Meeting functional requirements as described in the assignment description
- ♦Demonstrating a solid understanding of C++ concepts, including good practice

- ♦Demonstrating the ability to apply the C++ concepts covered in the unit to a large scale practical example
- ♦Following the unit Programming Style Guide
- ♦Creating solutions that are as efficient and extensible as possible

Late assignments will incur a 5% penalty per late day (including weekends), and may be submitted up to a maximum of 7 days late. After this time submissions will not be accepted without prior arrangement with the unit leader.

Note that submitted code MUST compile, otherwise ZERO marks will be awarded.

Due date:

Tuesday 26 May 2015

Examinations

- **Examination 1**

Weighting:

60%

Length:

3 hours

Type (open/closed book):

Closed book

Electronic devices allowed in the exam:

None

Remarks:

A sample exam will be made available a month before the examination. Full exam revision will be covered in Week 12.

Learning resources

Monash Library Unit Reading List (if applicable to the unit)
http://readinglists.lib.monash.edu/index.html

Feedback to you

Types of feedback you can expect to receive in this unit are:

- Informal feedback on progress in labs/tutes
- Graded assignments with comments
- Interviews
- Solutions to tutes, labs and assignments

Extensions and penalties

Submission must be made by the due date otherwise penalties will be enforced.

You must negotiate any extensions formally with your campus unit leader via the in-semester special consideration process: http://www.monash.edu.au/exams/special-consideration.html

Returning assignments

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

Resubmission of assignments

Students may not resubmit assignments after the due date has passed.

Assignment submission

It is a University requirement (http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-academic-integrity-managing-pla for students to submit an assignment coversheet for each assessment item. Faculty Assignment coversheets can be found at http://www.infotech.monash.edu.au/resources/student/forms/. Please check with your Lecturer on the submission method for your assignment coversheet (e.g. attach a file to the online assignment submission, hand-in a hard copy, or use an electronic submission). Please note that it is your responsibility to retain copies of your assessments.

Online submission

If Electronic Submission has been approved for your unit, please submit your work via the learning system for this unit, which you can access via links in the my.monash portal.

Recommended Resources

This unit will require the use of a personal computer and a suitable IDE for C++ development. While Visual Studio for Windows will be used in the laboratory environment, any IDE is suitable for outside development. Copies of the Windows operating system and Visual Studio 2013 may be obtained free of charge from http://msdnaa.monash.edu.au/fit

Additional subject costs

As "clickers" will be used in the lectures to facilitate greater in-class engagement, students are strongly encouraged to purchase a Turning Point clicker from the Campus Bookstore or directly from the Australian Distributor. The lecturer will discuss and demonstrate this in the first week.

Other Information

Policies

Monash has educational policies, procedures and guidelines, which are designed to ensure that staff and students are aware of the University's academic standards, and to provide advice on how they might uphold them. You can find Monash's Education Policies at: www.policy.monash.edu.au/policy-bank/academic/education/index.html

Faculty resources and policies

Important student resources including Faculty policies are located at http://intranet.monash.edu.au/infotech/resources/students/

Graduate Attributes Policy

http://www.policy.monash.edu/policy-bank/academic/education/management/monash-graduate-attributes-policy.h

Student Charter

www.opq.monash.edu.au/ep/student-charter/monash-university-student-charter.html

Student services

The University provides many different kinds of support services for you. Contact your tutor if you need advice and see the range of services available at http://www.monash.edu.au/students. For Malaysia see http://www.monash.edu.my/Student-services, and for South Africa see http://www.monash.ac.za/current/.

Monash University Library

The Monash University Library provides a range of services, resources and programs that enable you to save time and be more effective in your learning and research. Go to www.lib.monash.edu.au or the library tab in my.monash portal for more information. At Malaysia, visit the Library and Learning Commons at http://www.lib.monash.edu.my/. At South Africa visit http://www.lib.monash.ac.za/.

Disability Liaison Unit

Students who have a disability or medical condition are welcome to contact the Disability Liaison Unit to discuss academic support services. Disability Liaison Officers (DLOs) visit all Victorian campuses on a regular basis.

- Website: http://www.monash.edu/equity-diversity/disability/index.html
- Telephone: 03 9905 5704 to book an appointment with a DLO; or contact the Student Advisor, Student Commuity Services at 03 55146018 at Malaysia
- Email: dlu@monash.edu
- Drop In: Equity and Diversity Centre, Level 1, Building 55, Clayton Campus, or Student Community Services Department, Level 2, Building 2, Monash University, Malaysia Campus