



MONASH University
Information Technology

FIT1040
Programming fundamentals

Unit Guide

Semester 2, 2013

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: 01 Jul 2013

Table of Contents

<u>FIT1040 Programming fundamentals - Semester 2, 2013</u>	1
<u>Mode of Delivery</u>	1
<u>Contact Hours</u>	1
<u>Workload requirements</u>	1
<u>Unit Relationships</u>	1
<u>Prohibitions</u>	1
<u>Chief Examiner</u>	1
<u>Campus Lecturer</u>	1
<u>Clayton</u>	1
<u>Gippsland</u>	2
<u>Tutors</u>	2
<u>Clayton</u>	2
<u>Academic Overview</u>	3
<u>Learning Outcomes</u>	3
<u>Unit Schedule</u>	4
<u>Assessment Summary</u>	4
<u>Teaching Approach</u>	5
<u>Assessment Requirements</u>	6
<u>Assessment Policy</u>	6
<u>Assessment Tasks</u>	6
<u>Participation</u>	6
<u>Examinations</u>	7
<u>Examination 1</u>	7
<u>Learning resources</u>	8
<u>Reading list</u>	8
<u>Feedback to you</u>	8
<u>Extensions and penalties</u>	8
<u>Returning assignments</u>	9
<u>Assignment submission</u>	9
<u>Online submission</u>	9
<u>Recommended Resources</u>	9
<u>Other Information</u>	10
<u>Policies</u>	10
<u>Graduate Attributes Policy</u>	10
<u>Student services</u>	10
<u>Monash University Library</u>	10
<u>Disability Liaison Unit</u>	11
<u>Your feedback to Us</u>	11
<u>Other</u>	11

FIT1040 Programming fundamentals - Semester 2, 2013

This unit will provide students with an overview of the fundamentals required to create programs. Students will learn to develop descriptions of algorithms and program logic using pseudocode which will be implemented as working software programs using in a visual procedural programming language. The unit will explore a variety of application domains including: computer games, business and science applications, computer generated arts, computer-based simulations and the control of simple robots. The topics covered will include the fundamental concepts: data types and structures, basic types of input and output, program control structures, and modular design along with the basics of event-driven programming and objects. These topics will be covered while placing an emphasis on the need to design program code that is easy to maintain, readable, tested, and well documented.

At the end of the unit students will expected to be able to apply to knowledge and skills learned in further units that cover software development using industry strength programming languages.

Mode of Delivery

- Clayton (Day)
- Gippsland (Off-campus)

Contact Hours

2 hrs lectures/wk, 2 hrs laboratories/wk, 1 hr tutorial/wk

Workload requirements

You are expected to spend 12 hours per week on various activities including reading, communication with other students and unit lecturers, and preparation for learning tasks and formal assessments.

Unit Relationships

Prohibitions

FIT1002

Chief Examiner

Mr Peter O'Donnell

Campus Lecturer

Clayton

Stephen Huxford

Gippsland

Stephen Huxford

Tutors

Clayton

Rosemary Missier

Academic Overview

Learning Outcomes

At the completion of this unit students will have: An understanding of:

- the relationship between a problem description and program design;
- the use of design representations;
- the sequence of steps that a computer takes to translate source code into executable code; and
- primitive data types and basic data structures.

Developed attitudes that enable them to:

- adopt a problem-solving approach;
- recognise the importance of programming and documentation conventions;
- act in accordance with best practice, industry standards and professional ethics.

Developed the skills to:

- use diagrams to design solutions for programming problems;
- apply problem solving strategies;
- use pseudo-code to design algorithms;
- create and test simple computer programs;
- analyse and debug existing programs; and
- write a test plan.

Demonstrated the communication skills necessary to:

- produce documentation for a program; and
- explain an existing program.

Unit Schedule

Week	Activities	Assessment
0	Students should read the unit guide and become familiar with the assessment requirements of the unit.	No formal assessment or activities are undertaken in week 0
1	Introduction to programming with Scribble	
2	Finding errors in programs: testing and debugging	Assessment task 3: Laboratory work and short tutorial tests is assessed weekly between Weeks 2 - 11
3	Using variables in programs	
4	Making decisions in programs	
5	Using loops	
6	Using loops to make art	
7	Using lists in loops	Assignment 1: Programming Basics, due Sunday Midnight
8	Searching and sorting algorithms	
9	Using abstraction to represent game play	
10	Making games using sprites and messages	
11	Software development and programming environments I	Assignment 2: Advanced Programming Application, due Sunday Midnight; Assessment task 3: Laboratory Work and Short Tutorial Tests end
12	Software development and programming environments II	Assignment 2: Interviews held
	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.

Assessment Summary

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

Assessment Task	Value	Due Date
Programming Basics	10%	Sunday Midnight (local time) Week 7
Advanced Programming Application	10%	Sunday Midnight Week 11; Interviews held in Week 12
Laboratory and short tutorial tests	20%	The end of the tutorial or laboratory session in Weeks 2 - 11 in which the work is performed.
Examination 1	60%	To be advised

Unit Schedule

Teaching Approach

Assessment Requirements

Assessment Policy

Faculty Policy - Unit Assessment Hurdles

(<http://www.infotech.monash.edu.au/resources/staff/edgov/policies/assessment-examinations/unit-assessment-hu>)

Academic Integrity - Please see the Demystifying Citing and Referencing tutorial at

<http://lib.monash.edu/tutorials/citing/>

Assessment Tasks

Participation

• Assessment task 1

Title:

Programming Basics

Description:

This assignment will require the development of a software application with Scribble that reacts to user interface events, taking input from a user and then performing a complex calculation. The task will require creating a programming solution to a calculation-based problem, creating software that reacts to an event, makes decisions with a IF-THEN-ELSE logic, stores user input in variables, and makes calculations using variables.

Weighting:

10%

Criteria for assessment:

Detailed assessment criteria will be provided on the unit web site along with full details of the assignment task.

1. The application must run correctly. Evidence of testing is required.
2. The application must meet the problem specification.
3. The application logic should be documented with a flowchart.
4. The application sprites and scripts should be constructed in a way that makes them easy to understand and maintain.
5. Students should be able to answer questions about their own work.

Due date:

Sunday Midnight (local time) Week 7

• Assessment task 2

Title:

Advanced Programming Application

Description:

This assignment will require the development of a software application with Scribble that implements a single-user game. The task will require creating an abstraction of the game state and changing that state as the user plays the game. This will require the creation of appropriate data structures to store the game state, the player's movers, and scripts that represent the game rules. The software will make decisions with a IF-THEN-ELSE logic, process data using loop-based logic, and display appropriate status updates as the game is played, won or lost.

Weighting:

Assessment Requirements

10%

Criteria for assessment:

Detailed assessment criteria will be provided on the unit web site along with full details of the assignment task.

1. The application must run correctly. Evidence of testing is required.
2. The application must meet the problem specification.
3. The application logic should be documented with a flowchart.
4. The application sprites and scripts should be constructed in a way that makes them easy to understand and maintain.
5. Students should be able to answer questions about their own work during an interview scheduled outside the lab class.

Due date:

Sunday Midnight Week 11; Interviews held in Week 12

• **Assessment task 3**

Title:

Laboratory and short tutorial tests

Description:

In Weeks 2 - 11 students will be expected to write and execute code to perform a task specified at the start of their Laboratory session. The specified coding task will come from a Laboratory task specification sheet released prior to each Laboratory session allowing for preparation.

Students will be expected to complete a very short test at the end of each Tutorial session. The test will ask 5 short answer questions on the lecture material for the week the Tutorial is based on.

Weighting:

20%

Criteria for assessment:

Laboratory work will be assessed during the Laboratory session. Full marks will require both working code and good coding style with the latter carrying more weight.

Tutors will mark tutorial short tests after the tutorial. The questions will examine both conceptual and practical knowledge covered in the lecture slides relevant to the tutorial. Required answers will be short and generally attract a mark of 0 or 1 (scaled appropriately for inclusion into the final mark for the unit).

Due date:

The end of the tutorial or laboratory session in Weeks 2 - 11 in which the work is performed.

Examinations

• **Examination 1**

Weighting:

60%

Length:

3 hours

Type (open/closed book):

Closed book

Electronic devices allowed in the exam:

Learning resources

Reading list

Armoni, M. and M. Ben-Ari (2013) *Computer Science Concepts in Scratch*. Weizmann Institute of Science. [on-line] http://stwww.weizmann.ac.il/g-cs/scratch/scratch_en.html.

Gaddis, T. (2012) *Starting out with programming logic and design*. 3rd Edition. Addison Wesley.

Lane, A., B. Meyer and J. Mullins (2012) *Generative Art with Scribble: A Project Based Introduction to Programming*. Monash BlockBooks Series. [on-line] via Apple iTunes Store and <http://monash-blockbooks.appspot.com>

Lane, A., B. Meyer and J. Mullins (2012) *Simulation with Cellular: A Project Based Introduction to Programming*. Monash BlockBooks Series. [on-line] via Apple iTunes Store and <http://monash-blockbooks.appspot.com>

Lane, A., B. Meyer and J. Mullins (2012) *Robotics with Enchanting and LEGO® NXT: A Project Based Introduction to Programming*. Monash BlockBooks Series. [on-line] via Apple iTunes Store and <http://monash-blockbooks.appspot.com>

Sprankle, M & J. Hubbard (2012) *Problem solving and programming concepts*. 9th Edition. Prentice Hall.

Venit, S. & E. Drake (2011) *Prelude to programming: Concepts and design*. 5th Edition. Addison Wesley.

Monash Library Unit Reading List
<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
- Test results and feedback
- Quiz results
- 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.

Assignment submission

It is a University requirement

(<http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-procedures.html>) 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 online quiz). 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

BYOB (Build Your Own Blocks), “Scribble” Edition. This free software development tool is available for MacOS, Windows. It is available for download at: <http://monofonik.github.com/scribble/>

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

Key educational policies include:

- Academic integrity;
<http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-academic-integrity-policy.html>
- Assessment in Coursework Programs;
<http://www.policy.monash.edu/policy-bank/academic/education/assessment/assessment-in-coursework-policy.html>
- Special Consideration;
<http://www.policy.monash.edu/policy-bank/academic/education/assessment/special-consideration-policy.html>
- Grading Scale;
<http://www.policy.monash.edu/policy-bank/academic/education/assessment/grading-scale-policy.html>
- Discipline: Student Policy;
<http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-discipline-policy.html>
- Academic Calendar and Semesters; <http://www.monash.edu.au/students/dates/>
- Orientation and Transition; <http://intranet.monash.edu.au/infotech/resources/students/orientation/>
- Academic and Administrative Complaints and Grievances Policy;
<http://www.policy.monash.edu/policy-bank/academic/education/management/complaints-grievance-policy.html>
- Code of Practice for Teaching and Learning;
<http://www.policy.monash.edu.au/policy-bank/academic/education/conduct/suppdocs/code-of-practice-teaching-and-learning.html>

Graduate Attributes Policy

<http://www.policy.monash.edu/policy-bank/academic/education/management/monash-graduate-attributes-policy.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 Sunway 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 Sunway, 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 Community Services at 03 55146018 at Sunway 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, Sunway Campus

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

Other

Recognition of Prior Learning

Prior to the start of semester, students who have advanced programming skills are invited to attempt an on-line based assessment of their existing skills, knowledge and ability. Students who obtain a pass grade may choose to enroll in a more advanced programming unit in place of FIT1040 and receive an exemption (but not credit) for FIT1040.

The on-line test can be found at <http://dsslab.infotech.monash.edu.au:8080/moodle/>