



MONASH University
Information Technology

FIT3140
Advanced programming

Unit Guide

Semester 2, 2015

Copyright © Monash University 2014. All rights reserved. Except as provided in the Copyright Act 1968, this work may not be reproduced in any form without the written permission of the host Faculty and School/Department.

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: 23 Jul 2015

Table of Contents

<u>FIT3140 Advanced programming - Semester 2, 2015</u>	1
<u>Mode of Delivery</u>	1
<u>Workload Requirements</u>	1
<u>Additional workload requirements</u>	1
<u>Unit Relationships</u>	1
<u>Prerequisites</u>	1
<u>Chief Examiner</u>	2
<u>Campus Lecturer</u>	2
<u>Malaysia</u>	2
<u>Your feedback to Us</u>	2
<u>Previous Student Evaluations of this Unit</u>	2
<u>Academic Overview</u>	3
<u>Learning Outcomes</u>	3
<u>Unit Schedule</u>	4
<u>Teaching Approach</u>	4
<u>Assessment Summary</u>	4
<u>Assessment Requirements</u>	6
<u>Assessment Policy</u>	6
<u>Assessment Tasks</u>	6
<u>Participation</u>	6
<u>Examinations</u>	8
<u>Examination 1</u>	9
<u>Learning resources</u>	9
<u>Reading list</u>	9
<u>Feedback to you</u>	9
<u>Extensions and penalties</u>	9
<u>Returning assignments</u>	9
<u>Resubmission of assignments</u>	9
<u>Referencing requirements</u>	10
<u>Assignment submission</u>	10
<u>Online submission</u>	10
<u>Recommended Resources</u>	10
<u>Recommended text(s)</u>	10
<u>Examination material or equipment</u>	10
<u>Other Information</u>	11
<u>Policies</u>	11
<u>Faculty resources and policies</u>	11
<u>Graduate Attributes Policy</u>	11
<u>Student Charter</u>	11
<u>Student services</u>	11
<u>Monash University Library</u>	11
<u>Disability Liaison Unit</u>	11

FIT3140 Advanced programming - Semester 2, 2015

This unit develops the students' ability to design, implement and maintain moderately complex, realistically-sized programs using an Agile software development methodology. It builds upon the basic programming techniques introduced in introductory programming unit and offers the first introduction to the implementation of more complex real-world programs. Examples of such systems include compilers and interpreters, simulations, visualisation tools, drawing packages, database systems, graphical games. Such systems may be implemented in the context of non-traditional computing environments such as smartphone "apps". The unit may offer students the opportunity to get acquainted with a second programming language within the procedural-object oriented paradigm, such as C++, Python or one of their cousins, depending on the scope of the project chosen in a particular semester.

The unit bridges between core programming knowledge and the large-scale software engineering context. It will emphasise the implementation and use of intermediate to advanced data structures (such as search trees, hash structures, graphs and graph algorithms etc.) and the embedding into an actual computing system (i.e. interacting with the O/S, networking components etc).

Mode of Delivery

Malaysia (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 3-hour laboratory

(b.) Additional requirements (all students):

- A minimum of 7 hours independent study per week for completing lab and assignment work, private study and revision

See also Unit timetable information

Additional workload requirements

Substantial parts of the project work must be completed in pairs. Students will need to schedule time to meet with their project partners outside scheduled classes.

Unit Relationships

Prerequisites

FIT1008 or FIT2085

Chief Examiner

Dr Robert Merkel

Campus Lecturer

Malaysia

Dr. Sylvester Olubolu Orimaye

Consultation hours: TBA

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

The unit received reasonably high SETU ratings last time it was run, and there were no major issues identified.

The major change to the unit this semester is due to changes in the prerequisite units, as students no longer learn Java in FIT1008. This has meant that we have switched to Python and Kivy as our tools this semester. The previous native Android toolchain was not popular with students, as it was difficult to use, slow, and buggy. While this will obviously change the specifics of the programming techniques learned this semester, the design and management aspects of the unit have not changed significantly.

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 should be able to:

1. analyse a proposed software project, taking into account client, product, and team characteristics, and determine whether an Agile development approach is suitable for the project;
2. design a moderately complex software system, using good object-oriented design. practices, and implement, test, and package this design for client deployment using a modern object-oriented development toolchain;
3. apply appropriate Agile practices to manage a moderately-sized development project in a small team;
4. identify and describe requirements, including non-functional and quality properties (such as efficiency and usability), for a software system, and evaluate design alternatives and their impacts on these properties;
5. apply unfamiliar technologies in a software system, such as new languages, toolkits, or development tools, without extensive specific instruction in these technologies;
6. reflect on a software product and the development process that created it, and identify opportunities for incrementally improving both.

Unit Schedule

Week	Activities	Assessment
0		No formal assessment or activities are undertaken in week 0
1	The problem domain	
2	Introduction to Agile development	Identify project partner for group work
3	Requirements gathering in Agile development	Assignment 1 due Monday 10 August 2015
4	Agile Scrum and Spiking	Assignment 2 due Monday 17 August 2015
5	Software Design and Modelling	
6	Software Design Quality	Assignment 3 due Monday 31 August 2015
7	Software Performance Analysis	
8	Scrum Practices	Assignment 4 due Monday 14 September 2015
9	Software testing and Test-Driven Development	Assignment 5 (initial submission) due Monday 28th September
10	Software Debugging, Logging & Refactoring	
11	Back of the Envelope Calculations	
12	User Interfaces	Assignment 5 (final submission) due Monday 19 October
	SWOT VAC	No formal assessment is undertaken during 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.

- **Laboratory-based classes**

This learning and teaching approach provides the opportunity for practical experimentation with approaches taught in lectures.

Assessment Summary

Examination (3 hours): 50%; In-semester assessment: 50%

Assessment Task

Value

Due Date

Unit Schedule

Assignment 1 - Introductory programming assignment (individual)	5%	Monday 10 August 2015
Assignment 2 - Vision statement and risk analysis (pairs)	5%	Monday 17 August 2015
Assignment 3 - Spiking (pairs)	10%	Monday 31 August 2015
Assignment 4 - Design and benchmarking (pairs)	5%	Monday 14 September 2015
Assignment 5 - Iterations (pairs)	25 %	Initial submission due Monday 28 September 2015, final submission due Monday 19 October
Examination 1	50%	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-hurdles>)

Academic Integrity - Please see resources and tutorials at

<http://www.monash.edu/library/skills/resources/tutorials/academic-integrity/>

Assessment Tasks

Participation

Students are expected to attend lectures and lab classes.

While lab exercises will not be marked, many of the labs will be used for project work in pairs, including assessment. As such, students will need to attend these.

Students will identify a project partner to work with for Assignments 2 through 5, who is enrolled in the same lab class, by the end of week 2. Students will work with their partner for subsequent in-semester assessment unless there are exceptional circumstances.

Assignments 2 through 5 will be submitted with the project partner. Pairs are expected to share the workload and will usually receive the same mark. Individual contributions will be documented, and if workloads are not appropriately shared over the semester marks will be adjusted accordingly.

• Assessment task 1

Title:

Assignment 1 - Introductory programming assignment (individual)

Description:

Students will complete an introductory programming assignment which introduces them to the general problem domain of future assignments.

Relevant learning outcomes: 2, 3, 4, 5

Weighting:

5%

Criteria for assessment:

- ◆ Delivery of required functionality.
- ◆ Quality of program design.
- ◆ Quality of coding practices.
- ◆ Layout and documentation.

Due date:

Monday 10 August 2015

• **Assessment task 2**

Title:

Assignment 2 - Vision statement and risk analysis (pairs)

Description:

Project pairs will write a brief, high-level description of the key requirements of the semester-long project which forms the basis for subsequent assessment tasks. They will also identify and analyse key risks to the successful completion of the project.

Relevant learning outcomes: 1, 3, 4.

Weighting:

5%

Criteria for assessment:

- ◆ Effectively capturing key functional requirements.
- ◆ Identification and description of key attributes of system context.
- ◆ Identification and analysis of risks
- ◆ Readability and presentation quality of document.

Due date:

Monday 17 August 2015

• **Assessment task 3**

Title:

Assignment 3 - Spiking (pairs)

Description:

Project pairs will conduct one or more "spiking" exercises, conducting exploratory/prototype coding activities to better characterise and mitigate identified risks to the project. Each individual spike will consist of some exploratory/proof-of-concept coding, and a report which documents the information gained. Students will conduct demonstrations in labs.

Relevant learning outcomes: 2, 3, 4, 5, 6

Weighting:

10%

Criteria for assessment:

Reporting, including:

- ◆ Risk description
 - ◆ Description of coding tasks
 - ◆ Clear reporting of outcome
 - ◆ Assessment of risk after spiking.
- ◆ Effectiveness of coding undertaken to mitigate risk.

Due date:

Monday 31 August 2015

• **Assessment task 4**

Title:

Assignment 4 - Design and benchmarking (pairs)

Description:

Project pairs will work together to write a document describing aspects of the architecture and design of their system. As well as structured text, the document will include UML

Assessment Requirements

diagrams. To inform their design, pairs will conduct "benchmarking" to measure the performance of alternative designs, and report the results in written form.

Relevant learning outcomes: 2, 4, 5, 6.

Weighting:

5%

Criteria for assessment:

- ◆ Appropriate use of standard notation to document design.
- ◆ Completeness of design.
- ◆ Quality of design.
- ◆ Rigorousness of benchmarking.
- ◆ Analysis of benchmarking results.
- ◆ Documenting results of benchmarking.

Due date:

Monday 14 September 2015

• Assessment task 5

Title:

Assignment 5 - Iterations (pairs)

Description:

Project pairs will complete two iterations of the active development phase of the project. They will deliver working and tested software implementing several "user stories" as developed in earlier assignments. In the first iteration (for initial submission) they will complete initial basic functionality; in the second iteration (for final submission), as well as adding additional functionality they will conduct performance tuning. The deliverables will also include release notes and user documentation as required. Students will also provide a short written report reflecting on their project work, including considering the quality of the deliverables, characterising their contribution, and analyzing the process by which they managed their project.

Relevant learning outcomes: 2, 3, 4, 5, 6

Weighting:

25 %

Criteria for assessment:

- ◆ Completeness of implementation.
- ◆ Quality of implementation.
- ◆ Appropriateness of functional testing.
- ◆ Rigorousness of performance tuning (after second iteration)
- ◆ Correct use of provided tools.
- ◆ Appropriate supply of documentation.
- ◆ Quality of reflective report

Due date:

Initial submission due Monday 28 September 2015, final submission due Monday 19 October

Examinations

Assessment Requirements

• Examination 1

Weighting:

50%

Length:

3 hours

Type (open/closed book):

Closed book

Electronic devices allowed in the exam:

None

Learning resources

Reading list

- The Scrum Primer. Free download: <http://www.scrumprimer.com/>
- Scrum and XP from the Trenches by Henrik Kniberg. Free download: <http://www.infoq.com/minibooks/scrum-xp-from-the-trenches>

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

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

No resubmission of assignments will be permitted.

Referencing requirements

Code based on algorithms or information from third-party sources (such as books or websites) must acknowledge these sources in comments, in sufficient detail for markers to find and check these sources. Some assignments will explicitly prohibit the use of this - if so, you must not use them.

Where third party material is used in written assignments, they should be cited. Students may use any of the referencing methods described in the Monash University Library's tutorial on referencing to do so:

<http://guides.lib.monash.edu/content.php?pid=88267&sid=656564>

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

Students will develop for Android using the Google SDK integrated with Eclipse. Eclipse will be provided in laboratory work, but students may choose to use any Android development setup that they prefer..

Students may choose to use their own laptops if they wish; however, they will be responsible for all technical support and ensuring that their code runs on the specified platform.

Recommended text(s)

Kent Beck, Cynthia Andres, Erich Gamma. (). *Extreme Programming explained*. (2nd Edition) Addison-Wesley Professional (ISBN: 978-0321278654).

Examination material or equipment

Students should refer to the unit website for details about the exam.

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 Community 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