



COIT20258 Software Engineering

Term 2 - 2017

Profile information current as at 30/04/2024 02:38 am

All details in this unit profile for COIT20258 have been officially approved by CQUniversity and represent a learning partnership between the University and you (our student). The information will not be changed unless absolutely necessary and any change will be clearly indicated by an approved correction included in the profile.

General Information

Overview

This unit introduces you to the practical issues involved in the design and implementation of modern software applications. From a design perspective, the use of design patterns as a means of reusing design models that are accepted best practices and UML as a mechanism for the modelling of software components are emphasised. Students are also provided with a grounding in the principles of object oriented design. The importance of documentation, requirements, traceability and testing will also be emphasised. Software design is discussed within the context of both plan driven software development and agile software development. Note: If you have undertaken COIT20226 Software Design and Implementation then this unit cannot be enrolled.

Details

Career Level: *Postgraduate*

Unit Level: *Level 9*

Credit Points: 6

Student Contribution Band: 8

Fraction of Full-Time Student Load: 0.125

Pre-requisites or Co-requisites

Pre-requisites: COIT20247 Database Design and Development, COIT20256 Data Structures and Algorithms, and COIT20248 Information Systems Analysis and Design

Important note: Students enrolled in a subsequent unit who failed their pre-requisite unit, should drop the subsequent unit before the census date or within 10 working days of Fail grade notification. Students who do not drop the unit in this timeframe cannot later drop the unit without academic and financial liability. See details in the [Assessment Policy and Procedure \(Higher Education Coursework\)](#).

Offerings For Term 2 - 2017

- Brisbane
- Distance
- Melbourne
- Rockhampton
- Sydney

Attendance Requirements

All on-campus students are expected to attend scheduled classes – in some units, these classes are identified as a mandatory (pass/fail) component and attendance is compulsory. International students, on a student visa, must maintain a full time study load and meet both attendance and academic progress requirements in each study period (satisfactory attendance for International students is defined as maintaining at least an 80% attendance record).

Website

[This unit has a website, within the Moodle system, which is available two weeks before the start of term. It is important that you visit your Moodle site throughout the term. Please visit Moodle for more information.](#)

Class and Assessment Overview

Recommended Student Time Commitment

Each 6-credit Postgraduate unit at CQUniversity requires an overall time commitment of an average of 12.5 hours of study per week, making a total of 150 hours for the unit.

Class Timetable

[Regional Campuses](#)

Bundaberg, Cairns, Emerald, Gladstone, Mackay, Rockhampton, Townsville

[Metropolitan Campuses](#)

Adelaide, Brisbane, Melbourne, Perth, Sydney

Assessment Overview

1. **Practical Assessment**

Weighting: 50%

2. **Examination**

Weighting: 50%

Assessment Grading

This is a graded unit: your overall grade will be calculated from the marks or grades for each assessment task, based on the relative weightings shown in the table above. You must obtain an overall mark for the unit of at least 50%, or an overall grade of 'pass' in order to pass the unit. If any 'pass/fail' tasks are shown in the table above they must also be completed successfully ('pass' grade). You must also meet any minimum mark requirements specified for a particular assessment task, as detailed in the 'assessment task' section (note that in some instances, the minimum mark for a task may be greater than 50%). Consult the [University's Grades and Results Policy](#) for more details of interim results and final grades.

CQUniversity Policies

All University policies are available on the [CQUniversity Policy site](#).

You may wish to view these policies:

- Grades and Results Policy
- Assessment Policy and Procedure (Higher Education Coursework)
- Review of Grade Procedure
- Student Academic Integrity Policy and Procedure
- Monitoring Academic Progress (MAP) Policy and Procedure – Domestic Students
- Monitoring Academic Progress (MAP) Policy and Procedure – International Students
- Student Refund and Credit Balance Policy and Procedure
- Student Feedback – Compliments and Complaints Policy and Procedure
- Information and Communications Technology Acceptable Use Policy and Procedure

This list is not an exhaustive list of all University policies. The full list of University policies are available on the [CQUniversity Policy site](#).

Previous Student Feedback

Feedback, Recommendations and Responses

Every unit is reviewed for enhancement each year. At the most recent review, the following staff and student feedback items were identified and recommendations were made.

Feedback from Have your say

Feedback

Assignment feedback

Recommendation

Students will be provided with more detailed feedback

Unit Learning Outcomes

On successful completion of this unit, you will be able to:

1. Apply user interface and object-oriented principles in the design of software systems.
2. Employ UML and design patterns in the design and documentation of software systems.
3. Use SQL effectively in a range of different design scenarios.
4. Critique the design alternatives available to a developer for an application.
5. Design and implement non-distributed, layered applications.
6. Exploit the software engineering life cycle support functionality provided by modern software development environments.

Australian Computer Society (ACS) recognises the Skills Framework for the Information Age (SFIA). SFIA is in use in over 100 countries and provides a widely used and consistent definition of ICT skills. SFIA is increasingly being used when developing job descriptions and role profiles.

ACS members can use the tool MySFIA to build a skills profile at

<https://www.acs.org.au/professionalrecognition/mysfia-b2c.html>

This unit contributes to the following workplace skills as defined by SFIA:

- Systems Design (DESN)
- System Integration (SINT)
- Programming/Software Development (PROG)
- Data Analysis (DTAN)
- Database/Repository Design (DBDS)
- Testing (TEST)
- Applications Support (ASUP)

Alignment of Learning Outcomes, Assessment and Graduate Attributes

 N/A Level	 Introductory Level	 Intermediate Level	 Graduate Level	 Professional Level	 Advanced Level
---	--	--	--	--	--

Alignment of Assessment Tasks to Learning Outcomes

Assessment Tasks	Learning Outcomes					
	1	2	3	4	5	6
1 - Practical Assessment - 50%	•	•	•	•	•	•
2 - Examination - 50%	•	•		•		

Alignment of Graduate Attributes to Learning Outcomes

Graduate Attributes	Learning Outcomes					
	1	2	3	4	5	6
1 - Knowledge	○	○	○	○	○	○
2 - Communication		○		○	○	
3 - Cognitive, technical and creative skills	○	○	○	○	○	○
4 - Research						
5 - Self-management						
6 - Ethical and Professional Responsibility					○	
7 - Leadership						
8 - Aboriginal and Torres Strait Islander Cultures						

Alignment of Assessment Tasks to Graduate Attributes

Assessment Tasks	Graduate Attributes							
	1	2	3	4	5	6	7	8
1 - Practical Assessment - 50%	○	○	○		○	○		
2 - Examination - 50%	○	○	○		○	○		

Textbooks and Resources

Textbooks

COIT20258

Prescribed

Software Engineering

Edition: 10th. (2016)

Authors: Sommerville, I.

Pearson

Harlow , Essex , England

ISBN: 1-292-09613-6

Binding: Paperback

[View textbooks at the CQUniversity Bookshop](#)

IT Resources

You will need access to the following IT resources:

- CQUniversity Student Email
- Internet
- Unit Website (Moodle)
- NetBeans IDE

Referencing Style

All submissions for this unit must use the referencing style: [Harvard \(author-date\)](#)

For further information, see the Assessment Tasks.

Teaching Contacts

Dennis Jarvis Unit Coordinator

d.jarvis@cqu.edu.au

Schedule

Week 1 - 10 Jul 2017

Module/Topic	Chapter	Events and Submissions/Topic
Introduction; NetBeans	Chapter 1; Online Material	

Week 2 - 17 Jul 2017

Module/Topic	Chapter	Events and Submissions/Topic
Software Processes; Case Study 1 - AddressBook: Model/View (MV)	Chapter 2; Online Material	

Week 3 - 24 Jul 2017

Module/Topic	Chapter	Events and Submissions/Topic
Agile Software Development; Case Study 1 - AddressBook: Model/View/Presenter (MVP)	Chapter 3; Online Material	

Week 4 - 31 Jul 2017

Module/Topic	Chapter	Events and Submissions/Topic
Requirements Engineering; System Modelling	Chapter 4, Chapter 5	

Week 5 - 07 Aug 2017		
Module/Topic	Chapter	Events and Submissions/Topic
Requirements Analysis;	Online Material;	
Vacation Week - 14 Aug 2017		
Module/Topic	Chapter	Events and Submissions/Topic
Week 6 - 21 Aug 2017		
Module/Topic	Chapter	Events and Submissions/Topic
Architectural Design; GUI, DB Design	Chapter 6; Online Material	Phase 1 due
Week 7 - 28 Aug 2017		
Module/Topic	Chapter	Events and Submissions/Topic
Design and Implementation; Design Documents	Chapter 7; Online Material	
Week 8 - 04 Sep 2017		
Module/Topic	Chapter	Events and Submissions/Topic
Design Principles; Case Study 1 - AddressBook: Model/View/Controller (MVC)	Online Material; Online Material	
Week 9 - 11 Sep 2017		
Module/Topic	Chapter	Events and Submissions/Topic
Software Testing; Case Study 1 - AddressBook: Critique	Chapter 8; Online Material	
Week 10 - 18 Sep 2017		
Module/Topic	Chapter	Events and Submissions/Topic
Software Evolution; Case Study 2 - Wumpus game: Specification	Chapter 9; Online Material	
Week 11 - 25 Sep 2017		
Module/Topic	Chapter	Events and Submissions/Topic
Quality Management; Case Study 2 - Wumpus game: Design	Chapter 24; Online Material	Phase 2 due
Week 12 - 02 Oct 2017		
Module/Topic	Chapter	Events and Submissions/Topic
Configuration Management; Review	Chapter 25; Online Material	
Review/Exam Week - 09 Oct 2017		
Module/Topic	Chapter	Events and Submissions/Topic
Exam Week - 16 Oct 2017		
Module/Topic	Chapter	Events and Submissions/Topic

Assessment Tasks

1 Practical Assessment

Assessment Type

Practical Assessment

Task Description

This assignment consists of two phases.

Phase 1: Students will implement a Java application that conforms to the MVP (Model View Presenter) pattern. The view will contain a Swing based GUI and the model will interact with a Java DB database. The specification of the system to be

built (including the design document) will be provided on the unit web site.

Phase 2: Students will implement a revised specification for the Phase 1 system. Additional requirements will include the refactoring of the Phase 1 system to conform to the MVC (Model View Controller) pattern. The specification of the system to be built (including the design document) will be provided on the unit web site.

Assessment Due Date

Phase 1: Friday 25-Aug-2017 11:45 PM AEST; Phase 2: Friday 29-Sep-2017 11:45 PM AEST

Return Date to Students

Two weeks after submission

Weighting

50%

Assessment Criteria

Phase 1: Initial Implementation (Total = 20 marks)

Criteria	Marks Available
Application class	2
Model + interface classes	3
View + interface classes	3
Presenter class	3
Layer interactions (Application/MVP, V/P, P/M, M/database)	4
Source code documentation / variable naming / code layout	2
Package structure	1
Acceptance tests (0 if any test fails)	2

Phase 2: Revised Implementation (Total = 30 marks)

Criteria	Marks Available
Application class	4
Model + interface classes	4
View + interface classes	4
Controller class	4
Layer interactions (Application/MVC, V/C, C/M, M/V, M/database)	8
Source code documentation / variable naming / code layout	2
Package structure	2
Acceptance tests (0 if any test fails)	2

Referencing Style

- [Harvard \(author-date\)](#)

Submission

Online

Learning Outcomes Assessed

- Apply user interface and object-oriented principles in the design of software systems.
- Employ UML and design patterns in the design and documentation of software systems.
- Use SQL effectively in a range of different design scenarios.
- Critique the design alternatives available to a developer for an application.
- Design and implement non-distributed, layered applications.
- Exploit the software engineering life cycle support functionality provided by modern software development environments.

Graduate Attributes

- Knowledge

- Communication
- Cognitive, technical and creative skills
- Self-management
- Ethical and Professional Responsibility

Examination

Outline

Complete an invigilated examination.

Date

During the examination period at a CQUniversity examination centre.

Weighting

50%

Length

180 minutes

Exam Conditions

Open Book.

Materials

Calculator - all non-communicable calculators, including scientific, programmable and graphics calculators are authorised

Dictionary - non-electronic, concise, direct translation only (dictionary must not contain any notes or comments).

Academic Integrity Statement

As a CQUniversity student you are expected to act honestly in all aspects of your academic work.

Any assessable work undertaken or submitted for review or assessment must be your own work. Assessable work is any type of work you do to meet the assessment requirements in the unit, including draft work submitted for review and feedback and final work to be assessed.

When you use the ideas, words or data of others in your assessment, you must thoroughly and clearly acknowledge the source of this information by using the correct referencing style for your unit. Using others' work without proper acknowledgement may be considered a form of intellectual dishonesty.

Participating honestly, respectfully, responsibly, and fairly in your university study ensures the CQUniversity qualification you earn will be valued as a true indication of your individual academic achievement and will continue to receive the respect and recognition it deserves.

As a student, you are responsible for reading and following CQUniversity's policies, including the [Student Academic Integrity Policy and Procedure](#). This policy sets out CQUniversity's expectations of you to act with integrity, examples of academic integrity breaches to avoid, the processes used to address alleged breaches of academic integrity, and potential penalties.

What is a breach of academic integrity?

A breach of academic integrity includes but is not limited to plagiarism, self-plagiarism, collusion, cheating, contract cheating, and academic misconduct. The Student Academic Integrity Policy and Procedure defines what these terms mean and gives examples.

Why is academic integrity important?

A breach of academic integrity may result in one or more penalties, including suspension or even expulsion from the University. It can also have negative implications for student visas and future enrolment at CQUniversity or elsewhere. Students who engage in contract cheating also risk being blackmailed by contract cheating services.

Where can I get assistance?

For academic advice and guidance, the [Academic Learning Centre \(ALC\)](#) can support you in becoming confident in completing assessments with integrity and of high standard.

What can you do to act with integrity?



Be Honest

If your assessment task is done by someone else, it would be dishonest of you to claim it as your own



Seek Help

If you are not sure about how to cite or reference in essays, reports etc, then seek help from your lecturer, the library or the Academic Learning Centre (ALC)



Produce Original Work

Originality comes from your ability to read widely, think critically, and apply your gained knowledge to address a question or problem