COIT20245 *Introduction to Programming*
Term 3 - 2023

All details in this unit profile for COIT20245 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

In this unit, you will learn how to program using the Java programming language. It is assumed that you have little or no programming experience so you will be guided through the basics of application development using classes and objects. You will learn about the parts of a program including variables, types, and methods, and learn how to take input and produce output. A key aspect of this unit is practical, hands-on, simple application development and testing which you will do in an industry standard integrated development environment (IDE).

Details

Career Level: *Postgraduate*
Unit Level: *Level 8*
Credit Points: *6*
Student Contribution Band: *8*
Fraction of Full-Time Student Load: *0.125*

Pre-requisites or Co-requisites

Anti-requisite: COIT29222 Programming Principles.
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 3 - 2023

- Brisbane
- Melbourne
- Online
- 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: 20%
2. **Practical Assessment**
Weighting: 30%
3. **Project (applied)**
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 Unit coordinator reflection.

Feedback

Attendance particularly in lectures has been poor.

Recommendation

Possibly automate attendance recording in large lectures. Contact students with poor attendance and identify them as being at risk.

Feedback from Unit coordinator reflection.

Feedback

Academic misconduct has been a problem especially collusion.

Recommendation

Continue to educate students on the consequences of academic misconduct. Possibly employ third party software to detect such breaches. With such a large cohort it would be difficult to individualise the various assessment items.

Feedback from Student feedback.

Feedback

Some students find the unit's advanced level challenging, particularly when Java is used as the primary programming language, which may not align with the interests of all students in future programming pursuits.

Recommendation

Consider introducing Python as the primary teaching tool, as it could provide a more approachable and engaging experience for students navigating the complexities of an introductory unit with advanced content in Java.

Feedback from Student feedback.

Feedback

Students are happy with the teaching staff.

Recommendation

Continue employing experienced and dedicated teaching staff.

Unit Learning Outcomes

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

1. Develop professionally documented and thoroughly tested object-oriented applications using an industry standard integrated development environment (IDE)
2. Apply procedural concepts (methods, iteration, selection) to the realisation of object behaviour
3. Implement standard algorithms such as searching, sorting, and sequential processing for arrays and lists of objects
4. Employ the stream abstraction to process records contained in sequential text files
5. Demonstrate command of the subset of the programming language presented in this unit, including its syntax, type system, scope rules, and libraries.

The 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. The SFIA code is included:

- Programming/Software Development (PROG)
- Testing (TEST)

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
1 - Practical Assessment - 20%		●			
2 - Practical Assessment - 30%	●	●	●		
3 - Project (applied) - 50%			●	●	●

Alignment of Graduate Attributes to Learning Outcomes

Graduate Attributes	Learning Outcomes				
	1	2	3	4	5
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 - 20%	○	○	○		○			
2 - Practical Assessment - 30%	○	○	○		○			
3 - Project (applied) - 50%	○	○	○					

Textbooks and Resources

Textbooks
COIT20245

Prescribed

Java How to Program : Early Objects Edition
11th Edition (2018)
Authors: Paul Deitel and Harvey Deitel
Pearson Education
Upper Saddle River , NJ , USA
ISBN: 9780134743356
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)
- JDK 17 , https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html
- Apache NetBeans 13, https://netbeans.apache.org/download/index.html

Referencing Style

All submissions for this unit must use the referencing style: Harvard (author-date)
For further information, see the Assessment Tasks.

Teaching Contacts

Mahbub Ahmed Unit Coordinator
m.ahmed@cqu.edu.au

Schedule

Week 1 - 06 Nov 2023		
Module/Topic	**Chapter**	**Events and Submissions/Topic**
Introduction to Computers, the Internet and Java	Chapter 1	
Week 2 - 13 Nov 2023		
Module/Topic	**Chapter**	**Events and Submissions/Topic**
Introduction to Java Applications; Input/Output and Operators	Chapter 2	
Week 3 - 20 Nov 2023		
Module/Topic	**Chapter**	**Events and Submissions/Topic**
Control Statements: Part 1	Chapter 4	
Week 4 - 27 Nov 2023		
Module/Topic	**Chapter**	**Events and Submissions/Topic**
Control Statements: Part 2	Chapter 5	

Vacation Week - 04 Dec 2023

Module/Topic	Chapter	Events and Submissions/Topic

Week 5 - 11 Dec 2023

Module/Topic	Chapter	Events and Submissions/Topic
Introduction to Classes, Objects, Methods and Strings	Chapter 3	

Week 6 - 18 Dec 2023

Module/Topic	Chapter	Events and Submissions/Topic
Methods: A Deeper Look	Chapter 6	**Assignment One** Due: Week 6 Friday (22 Dec 2023) 11:59 pm AEST

Vacation Week - 25 Dec 2023

Module/Topic	Chapter	Events and Submissions/Topic

Week 7 - 01 Jan 2024

Module/Topic	Chapter	Events and Submissions/Topic
Arrays and ArrayLists : Part 1	Chapter 7: 7.1-7.7, 7.16	

Week 8 - 08 Jan 2024

Module/Topic	Chapter	Events and Submissions/Topic
Searching, Sorting and Big O	Chapter 19	

Week 9 - 15 Jan 2024

Module/Topic	Chapter	Events and Submissions/Topic
Arrays and ArrayLists: Part 2 (Continued from Week 7)	Chapter 7: 7.8 - 7.15	

Week 10 - 22 Jan 2024

Module/Topic	Chapter	Events and Submissions/Topic
Files and Streams; StringBuilder; Text Blocks; Design (intro)	Chapter 15; Chapter 14: 14.4; Online resources	**Assignment two** Due: Week 10 Friday (26 Jan 2024) 11:59 pm AEST

Week 11 - 29 Jan 2024

Module/Topic	Chapter	Events and Submissions/Topic
Strings, Characters and Regular Expressions	Chapter 14	

Week 12 - 05 Feb 2024

Module/Topic	Chapter	Events and Submissions/Topic
Revision and Complete Project	Lecture Notes from Weeks 1 - 12	

Exam Week - 12 Feb 2024

Module/Topic	Chapter	Events and Submissions/Topic
		Project Due: Exam Week Monday (12 Feb 2024) 11:59 pm AEST

Assessment Tasks

1 Assignment One

Assessment Type
Practical Assessment

Task Description
In this assignment you will use the key programming concepts and constructs studied in the first half of

the term to develop, test and document a software application.

See Moodle for the full specification of the assignment requirements.

Assessment Due Date
Week 6 Friday (22 Dec 2023) 11:59 pm AEST
Online via Moodle

Return Date to Students

Two weeks after submission

Weighting
20%

Assessment Criteria
The students will be assessed against their ability to:

1. use variables, constants, primitive and reference types, loops, selection statements, methods, input statements, output statements, objects and classes to develop a software application;
2. use an IDE (specifically NetBeans);
3. code according to standard (and good) coding practices;
4. develop a thorough test plan and to thoroughly test (and where necessary, debug) their code;
5. document an application;
6. follow a design specification.

More detailed marking criteria can be accessed from Moodle.

Referencing Style

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

Submission
Online

Learning Outcomes Assessed

- Apply procedural concepts (methods, iteration, selection) to the realisation of object behaviour

Graduate Attributes

- Knowledge
- Communication
- Cognitive, technical and creative skills
- Self-management

2 Assignment two

Assessment Type
Practical Assessment

Task Description
In this assignment you will use key constructs and concepts studied in the second part of the term (e.g. arrays, array lists, searching, sorting etc.) to develop, test and document a second software application. The application will also require the use of programming constructs studied in the first part of the term.

See Moodle for the full specification of the assignment requirements.

Assessment Due Date
Week 10 Friday (26 Jan 2024) 11:59 pm AEST
Online via Moodle

Return Date to Students

Two weeks after submission

Weighting
30%

Assessment Criteria

The students will be assessed against their ability to:

1. develop an application that uses classes, objects, standard searching and sorting algorithms, arrays, array lists and performs data validation and error handling as specified;
2. use an IDE (specifically NetBeans);
3. code according to standard (and good) coding practice;
4. develop a thorough test plan and to thoroughly test (and where necessary, debug) their code;
5. document an application;
6. follow a design specification.

More detailed marking criteria can be accessed from Moodle.

Referencing Style

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

Submission
Online

Learning Outcomes Assessed

- Develop professionally documented and thoroughly tested object-oriented applications using an industry standard integrated development environment (IDE)
- Apply procedural concepts (methods, iteration, selection) to the realisation of object behaviour
- Implement standard algorithms such as searching, sorting, and sequential processing for arrays and lists of objects

Graduate Attributes

- Knowledge
- Communication
- Cognitive, technical and creative skills
- Self-management

3 Project

Assessment Type
Project (applied)

Task Description
This assessment item will require students to develop a final project. It will involve aspects of design, software development and creation of associated program documentation. It will require the use of the programming concepts and constructs studied in weeks 1 to 11. Details can be found in the unit Moodle site.

Assessment Due Date
Exam Week Monday (12 Feb 2024) 11:59 pm AEST
Online via Moodle

Return Date to Students

Certification of grades

Weighting
50%

Assessment Criteria
The project will be assessed on quality, appropriateness, and correctness of design, as well as ability to:

- develop an application that uses the key programming constructs and concepts studied during the term;
- use an IDE (specifically NetBeans);
- code according to standard (and good) coding practice;
- develop a thorough test plan and to thoroughly test (and where necessary, debug) their code;
- document an application; and
- follow a design specification.

More detailed marking criteria can be accessed from Moodle.

Referencing Style

- Harvard (author-date)

Submission
Online

Learning Outcomes Assessed

- Implement standard algorithms such as searching, sorting, and sequential processing for arrays and lists of objects
- Employ the stream abstraction to process records contained in sequential text files
- Demonstrate command of the subset of the programming language presented in this unit, including its syntax, type system, scope rules, and libraries.

Graduate Attributes

- Knowledge
- Communication
- Cognitive, technical and creative skills

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