COIT20256 *Data Structures and Algorithms*
Term 3 - 2020

Profile information current as at 27/04/2024 04:48 am

General Information

Overview

In this unit, you will study advanced data structures and algorithms for software development using an object-oriented programming language. You will learn how to design and build classes, throw exceptions, and extend a class using inheritance and polymorphism. You will practise these concepts and develop applications with front-end Graphical User Interface (GUI) components and backend databases using database programming. You will build software applications using complex data structures and Application Programming Interfaces (APIs). You will gain an understanding of basic algorithms, and learn to evaluate algorithmic performance and assess the correct use of different data structures. You will be introduced to functional programming using Lambdas and Streams. You will obtain hands-on experience using all the concepts by completing programming exercises.

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-requisite: COIT20245 Introduction to Programming
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 - 2020

- 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: 10%
3. **Practical Assessment**
Weighting: 20%
4. **Take Home Exam**
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 Student evaluation

Feedback
This unit is well written and the assessments are challenging but extremely helpful in learning.

Recommendation
Continue with the current assessments that help students to develop practical skills.

Feedback from Student evaluation

Feedback
Expects more focus on data structures and algorithms.

Recommendation
Database programming and Object-oriented design will be replaced with graph theory and additional tutorials on algorithm.

Feedback from Student evaluation

Feedback
Assignments are difficult for students migrating from non-STEM background.

Recommendation
Provide more support for students and explore the potential for peer mentoring.

Unit Learning Outcomes

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

1. Design classes which use inheritance, polymorphism, and exception handling
2. Develop multi-layered software solutions, focusing on data structures and algorithms
3. Integrate data sets using complex data structures such as linked lists, stacks, and queues
4. Evaluate performance of different algorithms in problem solving
5. Investigate socially innovative practices in software development
6. Create Lambda expressions and streams using functional programming.

Australian Computer Society (ACS) recognises the Skills Framework for the Information Age (SFIA). SFIA provides a consistent definition of ICT skills. SFIA is adopted by organisations, governments, and individuals in many countries and is increasingly 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 7. The SFIA code is included:

- Software Design (SWDN)
- System Integration and Build (SINT)
- Programming/Software Development (PROG)
- Data modelling and design (DTAN)
- Database Design (DBDS)
- Testing (TEST)
- User experience analysis (UNAN)
- User experience design (HCEV).

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 - 20%	•	•			•	
2 - Practical Assessment - 10%				•		
3 - Practical Assessment - 20%	•	•	•		•	•
4 - Take Home Exam - 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 - 20%	◦	◦	◦		◦			
2 - Practical Assessment - 10%	◦	◦	◦					
3 - Practical Assessment - 20%	◦	◦	◦	◦	◦	◦		
4 - Take Home Exam - 50%	◦	◦	◦					

Textbooks and Resources

Textbooks

COIT20256

Prescribed

Java How to Program, Late Objects, Global Edition (11e)
Global Edition (11e) (2019)
Authors: Paul Deitel and Harvey Deitel
Pearson Higher Ed US
USA
ISBN: 978-1292273730
Binding: eBook

Additional Textbook Information
If you prefer to study from a paper copy, they are available at the CQUni Bookshop here: http://bookshop.cqu.edu.au (search on the Unit code)

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 11 - OpenJDK
- Apache Netbeans IDE 11.3
- Scene Builder 11.0

Referencing Style

All submissions for this unit must use the referencing styles below:

- Harvard (author-date)
- American Psychological Association 7th Edition (APA 7th edition)

For further information, see the Assessment Tasks.

Teaching Contacts

Mary Tom Unit Coordinator
m.tom@cqu.edu.au

Schedule

Week 1 - 09 Nov 2020

Module/Topic	Chapter	Events and Submissions/Topic
Classes and Objects: A Deeper Look	8	

Week 2 - 16 Nov 2020

Module/Topic	Chapter	Events and Submissions/Topic
Inheritance	9	

Week 3 - 23 Nov 2020

Module/Topic	Chapter	Events and Submissions/Topic

Polymorphism and Interfaces	10	

Week 4 - 30 Nov 2020

Module/Topic	Chapter	Events and Submissions/Topic
JavaFX GUI and Event-Driven Programming	12 and 13	

Vacation Week - 07 Dec 2020

Module/Topic	Chapter	Events and Submissions/Topic

Week 5 - 14 Dec 2020

Module/Topic	Chapter	Events and Submissions/Topic
Exception Handling, Files, Streams and Serialization	11 and 15	

Week 6 - 21 Dec 2020

Module/Topic	Chapter	Events and Submissions/Topic
Recursion	18	

Vacation Week - 28 Dec 2020

Module/Topic	Chapter	Events and Submissions/Topic

Week 7 - 04 Jan 2021

Module/Topic	Chapter	Events and Submissions/Topic
Generic Collections: Lists, Sets, Maps, and Priority Queue	16	**Assignment 1** Due: Week 7 Monday (4 Jan 2021) 11:45 pm AEST

Week 8 - 11 Jan 2021

Module/Topic	Chapter	Events and Submissions/Topic
Lambdas and Aggregate Operations for Collection Streams	17	

Week 9 - 18 Jan 2021

Module/Topic	Chapter	Events and Submissions/Topic
Sorting and Algorithmic Efficiency	19	

Week 10 - 25 Jan 2021

Module/Topic	Chapter	Events and Submissions/Topic
Implementing Stacks, Queues and Binary Search Tree	21	

Week 11 - 01 Feb 2021

Module/Topic	Chapter	Events and Submissions/Topic
Generic classes and methods	20	**Assignment 2** Due: Week 11 Thursday (4 Feb 2021) 11:45 pm AEST

Week 12 - 08 Feb 2021

Module/Topic	Chapter	Events and Submissions/Topic
Graphs and applications		

Exam Week - 15 Feb 2021

Module/Topic	Chapter	Events and Submissions/Topic

Term Specific Information

Unit Coordinator
Dr Mary Tom
College of Information & Communications Technology
School of Engineering and Technology
Central Queensland University
Brisbane QLD 4000, Australia
Phone: +61 7 3295 1119
Email: m.tom@cqu.edu.au

Assessment Tasks

1 Assignment 1

Assessment Type
Practical Assessment

Task Description
In this assignment you will demonstrate your ability to analyse the given problem, model and design data structures using UML class diagrams, and develop a software solution applying the Object-Oriented programming concepts of classes, inheritance, and polymorphism. You will also design and develop a graphical user interface (GUI) for the software solution applying event-driven programming. This assessment task is to design, code, debug, and test a software application using the topics learnt in Weeks 1 - 5. Further details are in the Assignment 1 specification document available from the Unit website.

Assessment Due Date
Week 7 Monday (4 Jan 2021) 11:45 pm AEST

Return Date to Students
Week 9 Tuesday (19 Jan 2021)

Weighting
20%

Assessment Criteria

1. Analyse given case study, choose and design appropriate data structures for application development
2. Design, and develop software solutions with the focus of data structures and algorithms
3. Apply classes, inheritance, polymorphism, and exception handling
4. Demonstrate socially innovative practices in software development
5. Appropriate use of Graphical User Interface (GUI)
6. Effective use of good coding practices
7. Rigorous testing of software applications.

Referencing Style

- Harvard (author-date)
- American Psychological Association 7th Edition (APA 7th edition)

Submission
Online

Submission Instructions
Submit one .zip file containing the source code files (.java) and the report file (.doc). Do not submit the zipped project folder or compiled binaries (.class or .jar).

Learning Outcomes Assessed

- Design classes which use inheritance, polymorphism, and exception handling
- Develop multi-layered software solutions, focusing on data structures and algorithms
- Investigate socially innovative practices in software development

Graduate Attributes

- Knowledge

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

2 Class Activities

Assessment Type
Practical Assessment

Task Description
This assessment item is to be developed and submitted as part of your weekly workshop sessions. It consists of a series of 6 practicals to be completed in weeks 3 – 10 (inclusive), except Weeks 5 and 6. For on campus students this work is due in the weekly tutorial, not the due date shown in Moodle. The date given in Moodle is to cater for all scheduled classes. *No marks will be awarded for work submitted outside the workshop/tutorial class or for late submissions.*
On-campus students

1. This assessment task must be completed and submitted in your weekly tutorial as and when directed by your lecturer/tutor.
2. Marks may be deducted if your tutor is not satisfied with your progress or understanding of the work.

Online students (DST mode)

1. Submit your weekly work by the due date shown on the unit website for the corresponding week. *Late submissions will be awarded 0 marks.*
2. The unit coordinator is your tutor and may make arrangements to discuss your work with you.

Assessment Due Date

This task commences with an “in-class” submission in week 3 and continues with “in-class” weekly submissions until the end of week 10. There is a total of 6 weekly submissions. On-campus students must complete and submit the scheduled work in their weekly tutorial..

Return Date to Students

The weekly submissions will be marked and returned within 1 week of their submission.

Weighting
10%

Assessment Criteria
Practical questions included in the weekly workshop material will be used to assess your understanding of the topics taught in that week and your ability to apply those principles to the given scenario. This will include designing data structures, algorithms, or writing part of source code. You are awarded total 1.5 marks for each weekly submission except for Weeks 2 and 10 submission having 2 marks.

Referencing Style

- Harvard (author-date)
- American Psychological Association 7th Edition (APA 7th edition)

Submission
Online

Submission Instructions
Submit your source code files (.java) and any documentation files (.doc) in one zip file.

Learning Outcomes Assessed

- Evaluate performance of different algorithms in problem solving

Graduate Attributes

- Knowledge
- Communication
- Cognitive, technical and creative skills

3 Assignment 2

Assessment Type
Practical Assessment

Task Description
In this assignment, you are required to analyse the given problem, model and design the required data structures using UML class diagrams. Develop a software application using data structures such as linked lists, queues, and streams, and having a front-end interactive Graphical User Interface (GUI) employing event-driven programming. You should also apply one or more algorithms such as searching, sorting. You should also justify the choice of appropriate data structure and effective use of algorithms. This assessment task includes design, document, develop code, debug , and test a java application applying topics learnt in Weeks 1 - 10. Further details are in the Assignment 2 specification document available from the Moodle Unit website.

Assessment Due Date
Week 11 Thursday (4 Feb 2021) 11:45 pm AEST

Return Date to Students
Exam Week Tuesday (16 Feb 2021)

Weighting
20%

Assessment Criteria

1. Efficient object-oriented program design.
2. Design and implement appropriate data structures for application development
3. Evaluate a variety of data structures and algorithmic approaches
4. Effective use of good programming practice/techniques
5. Rigorous testing of software application
6. Demonstrate socially innovative practices in software development
7. Work collaboratively as part of a small team.

Referencing Style

- Harvard (author-date)
- American Psychological Association 7th Edition (APA 7th edition)

Submission
Online

Submission Instructions
Submit one zip file containing the source code files (.java) per group and the individual report file (.doc) by each member of the group. Do not submit the zipped project folder or compiled binaries(.class or .jar).

Learning Outcomes Assessed

- Design classes which use inheritance, polymorphism, and exception handling
- Develop multi-layered software solutions, focusing on data structures and algorithms
- Integrate data sets using complex data structures such as linked lists, stacks, and queues
- Investigate socially innovative practices in software development
- Create Lambda expressions and streams using functional programming.

Graduate Attributes

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

4 Take Home Examination

Assessment Type
Take Home Exam

Task Description
The take home exam will have questions designed to test topics covered in weeks 7-12 inclusive. The questions will include completing partially completed source code solution to programming problems, writing short programs to solve given questions, provide output to program, and find errors and correct errors in given source code. You will also be asked to explain the source code or solutions written by you and justify choice of data structures or algorithms.

Assessment Due Date

During the University examination period

Return Date to Students

Marks will be released on certification date

Weighting
50%

Assessment Criteria
Demonstrate deeper understanding of theoretical aspects learnt during the term.
Correct use of complex data structures in programming
Analyse given source code to find and correct errors
Apply algorithmic performance to measure program efficiency
Apply lambdas and streams operation

Referencing Style

- Harvard (author-date)
- American Psychological Association 7th Edition (APA 7th edition)

Submission
Online

Learning Outcomes Assessed

- Integrate data sets using complex data structures such as linked lists, stacks, and queues
- Evaluate performance of different algorithms in problem solving
- Create Lambda expressions and streams using functional programming.

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