UNIVERSITY OF CANBERRA

Software Technology 2

Unit code and version	7170.6
Unit offering option	212836
Study level	Level 2 - Undergraduate Intermediate Unit
Credit points	3
Faculty	Faculty of Science and Technology
Discipline	Academic Program Area - Technology
Unit offering details	Semester 1, 2023 , ON-CAMPUS , UC - Canberra, Bruce
Unit convener name and contact details	Robert Cox Room:6C22 Phone 6201 5230 Email Robert.Cox@canberra.edu.au
Administrative contact details	Student Central Building 1, Level B E: Student.Centre@canberra.edu.au T: 1300 301 727

Academic content

Unit description

The unit provides an in-depth study of the software construction process. Topics covered include the specification, modularisation and verification of abstract data types (ADTs); data structures including arrays, stacks, queues, lists, trees, sets, maps, hash tables and heaps; and algorithms including sort, search and recursion. For ADTs in general, the unit deals with operations, representation and algorithms, space and time efficiency, and appropriateness for different applications. This unit may be cotaught with 9073 Software Technology 2 G.

Learning outcomes

On completion of this unit students will be able to:

1. Describe a variety of algorithm design techniques and recognise the principles of good software design;
2. Apply their knowledge of algorithms and data structures to writing efficient programs;

3. Analyse and compare commonly used algorithms and data structures;
4. Design efficient algorithms and choose appropriate data structures for real-world problems and explain their space and time efficiency; and
5. Evaluate existing software modules for function and purpose.

Graduate attributes

1. UC graduates are professional
    - use creativity, critical thinking, analysis and research skills to solve theoretical and real-world problems
2. UC graduates are global citizens
    - make creative use of technology in their learning and professional lives
3. UC graduates are lifelong learners
    - reflect on their own practice, updating and adapting their knowledge and skills for continual professional and academic development

Skills development

As students of the University of Canberra, you will develop your critical thinking skills, your ability to solve complex problems, your ability to work with others, your confidence to learn independently, your written communication skills, your spoken communication skills and a number of work-related knowledge and skills.

Prerequisites

4483 Software Technology 1.

Corequisites

None.

Accreditation

ACS Accreditation

This unit is part of courses accredited by the Australian Computer Society (ACS)

Skills Framework for the Information Age (SFIA)

This unit aligns with the following SFIA professional skills:

Programming/software development PROG

Methods and tools METL

Data modelling and design DTAN

Research RSCH

SFIA skills are defined by levels of responsibility, based on autonomy, influence, complexity, business skills, and knowledge. Although this unit may cover knowledge and skills at higher levels, it is expected that graduates of undergraduate degrees will be capable of operating at Level 2 overall.

Seoul Accord

The UC generic attributes address graduate attributes 1, 6, 7, 9, and 10 of the Seoul Accord. The remaining graduate attributes that are covered in this unit are:

2. Knowledge for Solving Computing Problems

3. Problem Analysis

4. Design/Development of Solutions

5. Modern Tool Usage

8. Computing Professionalism and Society

[The following is required for units required in ETB001 Bachelor of Engineering (Honours) and 354JA Master of Engineering – for other units, remove if unnecessary]

EA Accreditation

This unit is part of courses accredited by Engineers Australia (EA)

This unit assesses and exposes students to the following Professional Engineer Stage 1 Competencies:

1.1 Comprehensive, theory based understanding – Indicators Assessed: a

1.2 Conceptual understanding – Indicators Assessed: a

1.3 In-depth understanding – Indicators Assessed: a

1.4 Discernment – Indicators Exposed: a, b

1.5 Knowledge – Indicators Assessed: a, b, c; Exposed: d, e, f

1.6 Understanding – Indicators Assessed: a, d; Exposed b, c, e

2.1 Application of established engineering methods – Indicators Assessed: a, d, f, g, h, i; Exposed b, c, e

2.2 Fluent application of engineering techniques, tools and resources – Indicators Assessed: a, d, f, g, h, i; Exposed b, c, e, j

2.3 Application of systematic engineering synthesis and design processes – Indicators Assessed: a, d; Exposed b, c

2.4 Application of systematic approaches – Indicators Assessed: a, d, f; Exposed b, c, e

3.1 Ethical conduct and professional accountability – Indicators Assessed: a, d; Exposed b, c

3.2 Effective oral and written communication in professional and lay domains – Indicators Assessed: a, b

Timetable of activities

WEEK	TOPIC	
Week 1	Introduction to Java Introduction To Unit	
Week 2	Java and Big O notation First Tutorial	
Week 3	Java pointers More on Big O	
Week 4	Introduction to Data Structures	

	Introduction to Algorithms GUI programming	
Week 5	Algorithms: Recursion	
Week 6	Sorting	
Week 7	Searching	
Week 8	Class free period	
Week9	Collections, arrays and memory stacks and queues	
Week 10	Program Structure methods , function pointers, ADT, Polymorphsim, Interfaces, Multiple Inheritance	
Week 11	Object Design patterns	
Week 12	Modern languages and modern programming , Rust, Go, Scala and KotLin	
Week 13	Week 13 quiz	

Unit resources

Required texts

- There is no prescribed text book for this unit.
- Students should refer to the first weeks lecture on line or in person , and the associated lecture notes for ideas on books and resources they might find helpful

Materials and equipment

Computers and software tools I will use the BlueJ IDE with a recent Java version this will be supported in the Labs. The use of private personal computers and relevant software is beneficial, but not essential.

Students are advised to allocate several Gb of space on two USB drives (or cloud storage) for tutorial exercises and assignments.

Unit website

Each unit you are enrolled in has an online teaching site in the learning management system UCLearn. You access UCLearn through MyUC.

Social Media

Social media will be used as part of the teaching and learning of this unit, and relates to websites and applications that allow students and staff to create and share material, or interact via social networking activities. This can take many forms including text, images, audio, video, gestures (such as 'liking', 'favouriting', 'following') and other multimedia communications. Students and staff should be aware that social media creates an environment for limitless communication, collaboration, dialogue and information exchange but that there should be an awareness of the 'amplificatory' effect of popular content, which can portray both positive and negative outputs. Students should note that social media is not at the control of the University and, therefore, changes may be made to learning or assessment materials at late notice as a consequence of changes to content or access to the social media platform. Students are warned that there may be possible disturbing content that can be viewed when using social media, that is outside the control of the University. Students who are concerned with the use of social media for privacy or other issues are advised to speak with their unit convener as soon as possible.

For further information please refer to the section on Social Media in the Assessment Procedures.

Assessment

Assessment item details

Week 5 Quiz

Due date

Week 5

Weighting

10%

Assessment details

Multiple choice quiz on Canvas available for 24 hours but with a time limit.

Addresses learning outcomes

On completion of this unit students will be able to:

- 1. Describe a variety of algorithm design techniques and recognise the principles of good software design;
- 2. Apply their knowledge of algorithms and data structures to writing efficient programs;
- 3. Analyse and compare commonly used algorithms and data structures;

Related graduate attributes

1. UC graduates are professional
    - use creativity, critical thinking, analysis and research skills to solve theoretical and real-world problems
2. UC graduates are global citizens

- make creative use of technology in their learning and professional lives

3. UC graduates are lifelong learners
    - reflect on their own practice, updating and adapting their knowledge and skills for continual professional and academic development

Week 9 Quiz

Due date

Week 9

Weighting

10%

Additional information

The Week 9 and Week 13 Quizzes are 40% of the marks combined (hereafter referred to as the quiz marks) it is necessary to pass this component of the unit to pass the unit. Specifically, students need to get 20 of the 40 marks in these two quizzes combined to pass this unit.

Assessment details

Summative quiz covering all unit material to date, to be conducted on Canvas during the lecture time slot.

Addresses learning outcomes

On completion of this unit students will be able to:

- 1. Describe a variety of algorithm design techniques and recognise the principles of good software design;
- 2. Apply their knowledge of algorithms and data structures to writing efficient programs;
- 3. Analyse and compare commonly used algorithms and data structures;

Related graduate attributes

1. UC graduates are professional
    - use creativity, critical thinking, analysis and research skills to solve theoretical and real-world problems
2. UC graduates are global citizens
    - make creative use of technology in their learning and professional lives
3. UC graduates are lifelong learners
    - reflect on their own practice, updating and adapting their knowledge and skills for continual professional and academic development

Week 13 Quiz

Due date

Week 13

Weighting

30%

Additional information

The Week 9 and Week 13 Quizzes are 40% of the marks combined (hereafter referred to as the quiz marks) it is necessary

to pass this component of the unit to pass the unit. Specifically, students need to get 20 of the 40 marks in these two quizzes combined to pass this unit.

Assessment details

Summative quiz covering all unit material to date, to be conducted on Canvas during the lecture time slot.

Addresses learning outcomes

On completion of this unit students will be able to:

- 1. Describe a variety of algorithm design techniques and recognise the principles of good software design;
- 2. Apply their knowledge of algorithms and data structures to writing efficient programs;
- 3. Analyse and compare commonly used algorithms and data structures;

Related graduate attributes

1. UC graduates are professional
    - use creativity, critical thinking, analysis and research skills to solve theoretical and real-world problems
2. UC graduates are global citizens
    - make creative use of technology in their learning and professional lives
3. UC graduates are lifelong learners
    - reflect on their own practice, updating and adapting their knowledge and skills for continual professional and academic development

Assignment 1

Due date

Week 8

Weighting

20%

Assessment details

Programming assignment with short written component covering all material in the unit to date to be submitted on Canvas.

Addresses learning outcomes

On completion of this unit students will be able to:

- 1. Describe a variety of algorithm design techniques and recognise the principles of good software design;
- 2. Apply their knowledge of algorithms and data structures to writing efficient programs;
- 3. Analyse and compare commonly used algorithms and data structures;
- 4. Design efficient algorithms and choose appropriate data structures for real-world problems and explain their space and time efficiency; and
- 5. Evaluate existing software modules for function and purpose.

Related graduate attributes

1. UC graduates are professional
    - use creativity, critical thinking, analysis and research skills to solve theoretical and real-world problems
2. UC graduates are global citizens

- make creative use of technology in their learning and professional lives

3. UC graduates are lifelong learners
    - reflect on their own practice, updating and adapting their knowledge and skills for continual professional and academic development

Assignment 2

Due date

Week 13

Weighting

30%

Assessment details

Programming assignment with short written component covering all material in the unit to date to be submitted on Canvas.

Addresses learning outcomes

On completion of this unit students will be able to:

- 1. Describe a variety of algorithm design techniques and recognise the principles of good software design;
- 2. Apply their knowledge of algorithms and data structures to writing efficient programs;
- 3. Analyse and compare commonly used algorithms and data structures;
- 4. Design efficient algorithms and choose appropriate data structures for real-world problems and explain their space and time efficiency; and
- 5. Evaluate existing software modules for function and purpose.

Related graduate attributes

1. UC graduates are professional
    - use creativity, critical thinking, analysis and research skills to solve theoretical and real-world problems
2. UC graduates are global citizens
    - make creative use of technology in their learning and professional lives
3. UC graduates are lifelong learners
    - reflect on their own practice, updating and adapting their knowledge and skills for continual professional and academic development

Submission of assessment items

Responsibility for understanding

If there is any doubt with regard to the requirements of any particular assignments or assessment procedure, the onus for clarifying the issue rests with the student who should contact the unit convenor or tutor. Further, it is the responsibility of students to ensure that they are correctly enrolled in the unit and that the tutor and Student Administration have their correct contact details.

Full details for each assessment item will be available on the Canvas site.

Submission of java assignments will be made as a single file of 1 or more zipped up folders (using zip), and will need to contain the BlueJ project files, all source files, any test data files written by the student, along with a report and evidence of testing. Markers are not expected to convert from some other IDE.

Extensions

Students can apply for an extension to the submission due date for an assessment item due to extenuating, evidenced circumstances (specific details are found in the Assessment Procedures). An extension must be applied for before the due date. Documentary evidence (e.g. medical certificate) will be expected for an extension to be granted, however this will not guarantee that the application will be successful. The Unit Convener or relevant Program Director/Course Convener will decide whether to grant an extension and the length of the extension.

An Assignment Extension form is available from the Student Forms page.

Late submissions

The following late submission period and penalty is applicable to any teaching period commencing after 1 April 2024.

To support the provision of timely feedback to students within the unit, late penalties will apply for summative assessments where late submission is permitted. Late submissions without an approved extension or reasonable adjustment will result in a penalty of a mark reduction of 10% of the maximum available marks for the assessment item per day (or part thereof) up to and including three calendar days. If a student submits more than three calendar days late without an approved extension or reasonable adjustment, the student will be allocated a mark of zero for that assessment, with no feedback provided.

Approval of extensions based on extenuating circumstances will be dependent upon the production of supporting documentation and at the discretion of the unit convener.

For teaching periods commencing prior to 1 April 2024, a late penalty of 5 % of the maximum available marks for the assessment item per day (or part thereof) was applied up to and including seven calendar days. An assignment submitted over 7 days late will not be accepted.

Special assessment requirements

The Week 9 and Week 13 Quizzes are 40% of the marks combined (hereafter referred to as the quiz marks) it is necessary to pass this component of the unit to pass the unit. Specifically, students need to get 20 of the 40 marks in these two quizzes combined to pass this unit. It is also necessary to get 50% overall for the unit to pass the unit.

Supplementary assessment

Refer to the Assessment Policy and Assessment Procedures

Except as directed by University Rules suplementary assesment is not offered in this subject.

Academic integrity

Students have a responsibility to uphold University standards on ethical scholarship. Good scholarship involves building on the work of others and use of others' work must be acknowledged with proper attribution made. Cheating, plagiarism, and falsification of data are dishonest practices that contravene academic values. Refer to the University's Student Charter for more information.

To enhance understanding of academic integrity, all students are expected to complete the Academic Integrity Module (AIM) at least once during their course of study. You can access this module within UCLearn (Canvas) through the 'Academic Integrity and Avoiding Plagiarism' link in the Study Help site.

Use of Text-Matching Software

The University of Canberra uses text-matching software to help students and staff reduce plagiarism and improve understanding of academic integrity. The software matches submitted text in student assignments against material from various sources: the internet, published books and journals, and previously submitted student texts.

Assessment

Programming Assignments must be submitted as zip files, they should contain BlueJ Java project files, code files, reports and evidence of testing. Students should specificaly use ZIP, not some other archival method.

Student responsibility

Learner engagement

Activities	Estimated hours
Weekly Online Lecture on Canvas: 2 hours/week, 12 weeks	24
Weekly tutorial: 1 hours per tutorial, 12 weeks	12
Weekly study commitment, in addition to the 2 items above: 2 hours/week, 12 weeks	36
Assignments & Final Exam: 78 hours (6 hours/week, 13 weeks)	78
Total	150

Inclusion and engagement

It is strongly recommended that students who need assistance in undertaking the unit because of disability or an ongoing health condition register with the Inclusion and Engagement Office as soon as possible so that reasonable adjustment arrangements can be made.

Participation requirements

Your participation in both lecture and tutorial/lab activities will enhance your understanding of the unit content and therefore the quality of your assessment responses. Lack of participation may result in your inability to satisfactorily pass assessment items. The week 9 quiz and week 13 Quiz only run at the specifid times specifically in the lecture time slots in week 9 and week 13.

Withdrawal

If you are planning to withdraw please discuss with your Unit Convener. UC College students must also seek advice from the College.

Required IT skills

Common IT skills, such as writing a report electronically, using web browsers, and using LMS Systems (eg. Canvas) are required. It is assumed students are familiar with programming in Python or a similar programming language before commencing this course.

Work integrated learning

None

Student feedback

All students enrolled in this unit will have opportunities to provide anonymous feedback on the unit through the InterFace Student Experience Questionnaire (ISEQ). The request for your feedback will be posted on your InterFace page at least twice during a teaching period. InterFace can be accessed through MyUC.

Changes to unit based on student feedback

As a result of student feedback, the following changes have recently been made to the unit:

Teaching the unit using Java instead of C++

Authority of this unit outline

This unit outline must be read in conjunction with the University of Canberra's Policies and Procedures, including the Assessment Policy and associated Procedure. The Assessment Policy and Assessment Procedure include information on matters such as plagiarism, grade descriptors, moderation, feedback, and deferred exams.

Any change to the information contained in the Academic content and Assessment sections of this document, will only be made by the Unit Convener if the written agreement of the Program Director and a majority of students has been obtained; and if written advice of the change is then provided on the teaching site in UCLearn. If this is not possible, written advice of the change must be then forwarded to each student enrolled in the unit at their registered term address. Any individual student who believes themselves to be disadvantaged by a change is encouraged to discuss the matter with the Unit Convener.

Authority Text

Main

> Exception – Potential changes to a unit's learning activities and assessment items (Approved Academic Board 2020)
>
> In the event of Australian Government and/or ACT Government directive, such as those requiring physical distancing and restrictions on movement because of a pandemic, learning activities and/or assessment items in some units may change. These changes will not be updated in the published Unit Outline but will be communicated to students via the unit's UCLearn (Canvas) teaching site. The new learning activities and/or assessment items will continue to meet the unit's learning outcomes, as described in the Unit Outline.
>
> New learning activities and/or assessment items will be available on the unit's UCLearn (Canvas) teaching site. Please contact the Unit Convener with any questions.

University of Canberra, Bruce ACT 2617 Australia
+61 2 6201 5111
ABN 81 633 873 422
CRICOS 00212K
TEQSA Provider ID: PRV12003 (Australian University)

UC acknowledges the Ngunnawal people, traditional custodians of the lands where Bruce campus is situated. We wish to acknowledge and respect their continuing culture and the contribution they make to the life of Canberra and the region. We also acknowledge all other First Nations Peoples on whose lands we gather.