SUBJECT OUTLINE

48024 Applications Programming

Course area	UTS: Engineering
Delivery	Autumn 2022; City
Credit points	6cp
Requisite(s)	48023 Programming Fundamentals OR 31267 Programming Fundamentals OR 31465 Object-oriented Programming OR 48430 Fundamentals of C Programming OR 41039 Programming 1
Result type	Grade and marks

Attendance: 3hpw. Forms of attendance and mode of delivery in this subject have changed to enable social distancing and reduce the risks of spreading COVID-19 in our community.
Recommended studies: basic skills in Java and Python programming

Subject coordinator

Dr Angela Huo

Email: huan.huo@uts.edu.au

Phone: 9514 4620

Location: CB11.07.129

Questions regarding assessment or content within the subject are welcome in lectures or tutorials or alternatively post them to the discussion board on Canvas. This helps ensure that all students get the benefit of the answers given. Answers to common questions related to the subject content, assignments etc. can be found on the FAQs page.

Students are expected to seek help through the following steps:

- Step 1: Check the FAQ page
- Step 2: Ask peers via the group discussion board
- Step 3: Ask your tutor
- Step 4: Ask the subject coordinator

The Subject Coordinator may be contacted by email if you have matters of a personal nature to discuss, e.g., illness, study problems, and for issues to do with extensions, group problems or other matters of importance.

All emails sent to subject coordinators, tutors or lecturers must have a clear subject line that states the subject number followed by the subject of the email (e.g. [48024]Request for Extension), and must be sent from your UTS email address.

Consultation hours: Check the Canvas Contact section for details on consultation hours. Requests for appointments outside the given consultation hours may be arranged where circumstances require, and to do so please contact the subject coordinator by email.

Teaching staff

Dr Angela Huo

Email: huan.huo@uts.edu.au

Detailed teaching staff information is available on Canvas

Subject description

This subject teaches students how to design, develop and evaluate software systems to meet predefined quality characteristics of functionality (suitability) and usability (understandability, learnability, operability, compliance). Software solutions are implemented using Java or Python. Concepts, theories and technologies underlying the methods and techniques are introduced and explained as required. Students apply all that they have learned to develop and implement the architecture of a business system.

Subject learning objectives (SLOs)

Upon successful completion of this subject students should be able to:

1. Evaluate if a solution is well-designed and fit for purpose
2. Design a well-constructed OO solution from a specification.
3. Demonstrate a working knowledge of lists.
4. Use inheritance to improve the system design.
5. Construct a GUI interface from a specification.

Course intended learning outcomes (CILOs)

This subject also contributes specifically to the development of the following Course Intended Learning Outcomes (CILOs):

- Design Oriented: FEIT graduates apply problem solving, design and decision-making methodologies to develop components, systems and processes to meet specified requirements. (C.1)
- Technically Proficient: FEIT graduates apply abstraction, mathematics and discipline fundamentals, software, tools and techniques to evaluate, implement and operate systems. (D.1)

Contribution to the development of graduate attributes

Engineers Australia Stage 1 Competencies

This subject contributes to the development of the following Engineers Australia Stage 1 Competencies:

- 1.2. Conceptual understanding of the mathematics, numerical analysis, statistics, and computer and information sciences which underpin the engineering discipline.
- 2.2. Fluent application of engineering techniques, tools and resources.

Teaching and learning strategies

The subject has one contact hour in Week 1 and three contact hours per week in the remaining weeks.

A weekly online study module presents new material in the form of videos and written articles and shows working examples of code. Each study module on average is expected to take 1 hour to complete and needs to be completed before coming to the lecture and the lab. Note that access to the lab material is electronically blocked until the student has completed the required weekly study module.

Regular lectures include interactive quizzes and reflections on the key topics and techniques.

A weekly two-hour lab session gives students practice working in groups to analyse software specifications and construct software plans, and practice working individually to write, debug and run code.

Content (topics)

1. Basic programming plans
2. Design rules and notations
3. Debugging
4. Strings
5. Classes
6. Linked lists
7. Inheritance
8. How to build a GUI
9. The event model
10. GUI Lists and list models
11. GUI tables and table models

Program

Week/Session	Dates	Description

0	14 February	**Preparation and Activities before start:** 1. Check the enrolment in Canvas 2. Check the enrolment in Lab access 3. Check the enrolment in ED **Activities to be completed this week**: 1. Read the subject outline 2. Complete the self-assessment Java quiz 3. Complete the 'Java Refresher' activities as per the quiz results 4. Complete the 'Python Refresher'
1	21 February	**Topic**: Introduction to the subject **Learning Material**: 1.Week 1 module 2.Module 2 pre-class activities **Lab**: None; Labs will commence from week 2. **Notes:** Pre-class activities **MUST be completed before Lab class**, as you will be prevented from accessing the lab exercises until after the study module has been completed.
2	28 February	**Topic**: Basic patterns read, count, sum, read loop, output, max **Learning Material**: 1. Lab 2 2. Module 3 pre-class activities **Lab**: Working in small groups, devise pattern-based plans for a set of problems, then code them individually. **Notes:** Each Lab **MUST be completed before the due date**, as you will be prevented from accessing the next lab until after the lab has been completed.
3	7 March	**Topic**: Basic process key/framework incremental goals code, test and debug **Learning Material**: 1. Lab 3 2. Module 4 pre-class activities **Lab**: Working in small groups, apply the basic processes to solve a set of problems, then code them individually.

		Notes: Lab assessment 1
4	14 March	**Topic**: Methods and strings functions vs procedures string functions process: break it down, build it up patterns: read, merged read loop, any, none, every **Learning Material**: 1. Lab 4 2. Module 5 pre-class activities **Lab**: Working in small groups, break a large goal down into sub-goals, devise plans for each sub-goal, then code them individually. **Notes:** Lab assessment 2
5	21 March	**Topic**: Classes Design: encapsulation, push it right, spread plans across classes, hide by default Syntax: classes, fields, and constructors, creating and using objects Process: design from the words, class diagram **Learning Material**: 1. Lab 5 2. Module 6 pre-class activities **Lab**: A problem will be analyzed in a group discussion. Students will code the solution to the problem individually. **Notes:** Lab assessment 3 Assignment 1 released, worth 35%
6	28 March	**Topic**: Lists List concept List methods Patterns: lookup, match, menu **Learning Material**: 1. Lab 6 2. Module 7 pre-class activities **Lab**: A list-based problem will be analysed in a group discussion. Students will

		code the solution to the problem individually. **Notes:** Lab assessment 4
7	4 April	**Topic**: System design Syntax: Interfaces and superclasses Process review: putting it all together, location table **Learning Material**: 1. Lab 7 2. Module 8 pre-class activities **Lab**: You can polish your assignment. **Notes:** **Pre-lab assessment 5 in Module 7 must be completed before Lab 7.**
STUVAC	11 April	
8	18 April	**Topic**: GUIs and events JavaFX stages, scenes, boxes, grid panes and GUI components Buttons, events and event handlers Patterns: the observer pattern **Learning Material**: 1. Lab 8 2. Module 9 pre-class activities **Lab**: Build a set of small GUI programs. **Notes:** 18 April is a holiday. Lab assessment 6 Assignment 1 due at 11:59pm 19 April 2022.
9	25 April	**Topic**: The MVC architecture Model-View-Controller concept FXML, observable properties, property bindings Patterns: JavaFX property patterns **Learning Material**: 1. Lab 9 2. Module 10 pre-class activities

		Lab: You will code an MVC application following the same steps as the demonstration. **Notes:** 25 April is a holiday. Lab assessment 7 Assignment 2 released, worth 25% *
10	2 May	**Topic**: GUI lists List views, observable lists, populating a list, list selection Controls Opening and closing windows Packages **Learning Material**: 1. Lab 10 2. Module 11 pre-class activities **Lab**: You will extend your program from week 9 by adding GUI lists and multiple windows.
11	9 May	**Topic**: GUI tables Table views, populating a table, table selection Change listeners Exceptions **Learning Material**: 1. Lab 11 2. Module 12 pre-class activities **Lab**: You will extend your program from week 10 by adding GUI Tables. You can also polish your assignment. **Notes:** Assignment 2 due at 11:59 pm 17 May 2022.
12	16 May	**Topic**: Review and sample quiz **Lab**: Assignment demonstration **Notes:** **Attendance is mandatory for assignment demonstration in lab class.**

STUVAC	23 May	Preparation for LMS exam. **Notes:** Advanced challenges due at 11:59 pm 10 June 2022.

The weekly schedule may be adjusted during the session to suit the needs of the class.

Students are encouraged to attend all scheduled classes for this subject.

This subject has compulsory online study modules that you must complete before coming to your classes.

Additional information

Assignments in this subject should be your own original work. Similarly, any group work should be the result of collaboration only within the group.

For more detail, go to http://www.uts.edu.au/current-students/feit/study-and-assessment-resources/academic-integrity-plagiarism-and-cheating The University's rules regarding academic misconduct can be found at www.gsu.uts.edu.au/rules/16-2.html.

Assessment

All assessments are individual. You can discuss approaches with tutors or labmates, and help each other with debugging, but you should write every line of code you submit except for the code provided in the form of lecture notes, reference solutions, lab solutions, and demo solutions. All assessments are submitted to system. Assessments that have a text-based user interface are marked by system, and those that have a graphical user interface are marked by a combination of system and the tutor (Assignment 2 and Lab assessment 7).

Assessment task 1: Labs

Intent: The labs, both assessed and non-assessed, provide practice in programming.

Objective(s): This assessment task addresses the following subject learning objectives (SLOs):

1, 2 and 3

This assessment task contributes to the development of the following Course Intended Learning Outcomes (CILOs):

C.1 and D.1

Type: Laboratory/practical

Groupwork: Individual

Weight: 14%

Task: The task is to code a well-designed solution to a small specification.

The output of this task is a program of one or more classes.

Length: Each weekly lab on average should take about 2 hours for the typical student.

Due: See Further information.

Criteria: The text-based labs are marked by system, and the GUI labs are marked by a combination of system, and the tutor. The minimum essential requirement for a pass in text-based lab is to pass all the test cases.

Further information: **Release date:** Each lab is released at the beginning of the lab sessions. Students must also complete the prerequisite study module in order for the lab to be released.

Due date: The due date for each lab session is stated in each lab introduction page. A typical week's lab exercise is due 1 hour before the release of the following week's lab. If there is no scheduled class in the following week (e.g. due to StuVac), the lab will still be due 167 hours after the release date. No late lab submission will be assessed.

Assessment task 2: Programming assignment 1

Intent: This programming assignment provides practice and tests for the topics of system design and lists.

Objective(s): This assessment task addresses the following subject learning objectives (SLOs):

1, 2 and 3

This assessment task contributes to the development of the following Course Intended Learning Outcomes (CILOs):

C.1 and D.1

Type: Project

Groupwork: Individual

Weight: 35%

Task: The task is to develop a well-designed OO system that uses lists. The output of this task is a solution of one or more classes.

Due: 11.59pm Tuesday 19 April 2022

Criteria: The assignment is marked on correctness by system instantly, based on the match to the model solution’s IO trace. Design rules and spoofy check will be manually performed after due.

The minimum essential requirement for a pass in this assessment is to complete the tasks worth half the mark.

Assessment task 3: Programming assignment 2

Intent: This programming assignment provides practice and tests for the topics of system design, GUIs and MVC.

Objective(s): This assessment task addresses the following subject learning objectives (SLOs):

1, 2, 3, 4 and 5

This assessment task contributes to the development of the following Course Intended Learning Outcomes (CILOs):

C.1 and D.1

Type: Project

Groupwork: Individual

Weight: 25%

Task: The task is to develop a well-designed OO GUI system that uses the MVC framework.

The output of this task is a program of one or more classes.

Due: 11.59pm Tuesday 17 May 2022
See also Further information.

Criteria: The assignment is marked on correctness and design principles, which are marked by a combination of automated system marking, demonstration and the tutor.

The minimum essential requirement for a pass is to complete tasks worth more than half the marks.

Further information: If, due to extenuating circumstances, you are unable to attend the demo session, please contact your tutor or the subject coordinator before the assessment task is due to discuss an alternative solution. Absence from the demo session without approval is subject to a penalty of 50 per cent of the marks for the assignment.

Assessment task 4: Timed LMS exam

Intent: Students are required to complete an LMS exam within a time frame to consolidate the factual knowledge taught in the subject.

Objective(s): This assessment task addresses the following subject learning objectives (SLOs):

1, 2, 3, 4 and 5

This assessment task contributes to the development of the following Course Intended Learning Outcomes (CILOs):

C.1 and D.1

Type: Quiz/test

Groupwork: Individual

Weight: 20%

Task: Answer 20 multiple-choice and fill-in-blank questions. The quiz is open book but restricted from accessing external webpages and application programs.

Length: 30 minutes

Due: UTS Exam period

Criteria: The questions are marked for correctness only. Feedback and reflection in lectures and labs will be relevant to enhance student performance in the exam.

Assessment task 5: Advanced Challenges

Intent: The advanced challenges provide practice and tests for programming proficiency.

Objective(s):	This assessment task addresses the following subject learning objectives (SLOs): 1, 2, 3 and 4 This assessment task contributes to the development of the following Course Intended Learning Outcomes (CILOs): C.1 and D.1
Type:	Quiz/test
Groupwork:	Individual
Weight:	6%
Task:	The task is to code a well-designed solution to an interview level specification. The output of this task is a program of one or more classes.
Length:	30-200 lines of code
Due:	UTS Exam period
Criteria:	The questions are marked for correctness only.

Assessment feedback

Lab assessments 1-6, Assignment 1 and advanced challenges are marked by system, so you get feedback on the correctness as soon as you submit a solution; Lab assessment 7 and Assignment 2 use a GUI, so you cannot get feedback instantly. Ask your lab assistant for feedback on your lab solutions; you can also ask your lab assistant for detailed feedback on your assignment solution after the due date.

Minimum requirements

In order to pass the subject, a student must achieve an overall mark of 50% or more.

References

You should study the weekly module videos and sample programs posted to Canvas and ED.

Other resources

All subject material is posted in Canvas and ED system.

The following websites may be useful:

- https://docs.oracle.com/javase/8/docs/
- https://docs.oracle.com/javase/tutorial/
- http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/ui_controls.htm
- http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- https://netbeans.org/downloads/
- https://perso.ensta-paris.fr/~diam/java/online/notes-java/index.html

U:PASS

UTS Peer Assisted Study Success is a voluntary “study session” where you will be studying the subject with other students in a group. It is led by a student who has previously achieved a distinction or high distinction in the subject area, and who has a good WAM. Leaders will prepare activities for you to work on in groups based on the content you are learning in lectures and tutorials. It’s really relaxed, friendly, and informal. Because the leader is a student just like you, they understand what it’s like to study the subject and how to do well, and they can pass those tips along to you. Students also say it’s a great way to meet new people and a “guaranteed study hour”.

You can sign up for U:PASS sessions via U:PASS website https://www.uts.edu.au/current-students/support/upass/upass/sessions/register-sessions. Note that sign up is not open until week 2, as it's voluntary and only students who want to go should sign up.

If you have any questions or concerns about U:PASS, please contact Georgina at upass@uts.edu.au, or check out the website.

Graduate attribute development

For a full list of the faculty's graduate attributes refer to the FEIT Graduate Attributes webpage.

For the contribution of subjects taken in the Bachelor of Engineering (Honours) or Master of Professional Engineering to the Engineers Australia Stage 1 Competencies, see the faculty's Graduate Attributes and the Engineers Australia Stage 1 Competencies webpage.

Assessment: faculty procedures and advice

Marking criteria

Marking criteria for each assessment task is available on the Learning Management System: Canvas.

Extensions

When, due to extenuating circumstances, you are unable to submit or present an assessment task on time, please contact your subject coordinator before the assessment task is due to discuss an extension. Extensions may be granted up to a maximum of 5 days (120 hours). In all cases you should have extensions confirmed in writing.

Special consideration

If you believe your performance in an assessment item or exam has been adversely affected by circumstances beyond your control, such as a serious illness, loss or bereavement, hardship, trauma, or exceptional employment demands, you may be eligible to apply for Special Consideration.

Late penalty

For Graded subjects:

Work submitted late without an approved extension is subject to a late penalty of 10 per cent of the total available marks deducted per calendar day that the assessment is overdue (e.g. if an assignment is out of 40 marks, and is submitted (up to) 24 hours after the deadline without an extension, the student will have four marks deducted from their awarded mark). Work submitted after five calendar days is not accepted and a mark of zero is awarded.

For some assessment tasks a late penalty may not be appropriate – these are clearly indicated in the subject outline. Such assessments receive a mark of zero if not completed by/on the specified date. Examples include:

a. weekly online tests or laboratory work worth a small proportion of the subject mark, or
b. online quizzes where answers are released to students on completion, or
c. professional assessment tasks, where the intention is to create an authentic assessment that has an absolute submission date, or
d. take-home papers that are assessed during a defined time period, or
e. pass/fail assessment tasks.

For Pass/Fail subjects:

Work submitted late without an approved extension will only be assessed at the subject coordinator's discretion. Students who do not submit assessment tasks by the due dates may be referred to the Responsible Academic Officer under Student Rule 3.8.2, and a fail result may be recorded for the subject.

Querying results

If you believe an error may have been made in the calculation of your result in an assessment task or the final result for the subject, it is possible to query the result with the Subject Coordinator within five (5) working days of the date of release of the result.

Academic liaison officer

Academic liaison officers (ALOs) are academic staff in each faculty who assist students experiencing difficulties in their studies due to: disability and/or an ongoing health condition; carer responsibilities (e.g. being a primary carer for small children or a family member with a disability); and pregnancy.

ALOs are responsible for approving adjustments to assessment arrangements for students in these categories. Students who require adjustments due to disability and/or an ongoing health condition are requested to discuss their situation with an accessibility consultant at the Accessibility Service before speaking to the relevant ALO.

Statement about assessment procedures and advice

This subject outline must be read in conjunction with the Coursework Assessments policy and procedures.

Statement on copyright

Teaching materials and resources provided to you at UTS are protected by copyright. You are not permitted to re-use these for commercial purposes (including in kind benefit or gain) without permission of the copyright owner. Improper or illegal use of teaching materials may lead to prosecution for copyright infringement.

Statement on plagiarism

Plagiarism and academic integrity

At UTS, plagiarism is defined in Rule 16.2.1(4) as: 'taking and using someone else's ideas or manner of expressing them and passing them off as ... [their] own by failing to give appropriate acknowledgement of the source to seek to gain an advantage by unfair means'.

The definition infers that if a source is appropriately referenced, the student's work will meet the required academic standard. Plagiarism is a literary or an intellectual theft and is unacceptable both academically and professionally. It can take a number of forms including but not limited to:

- copying any section of text, no matter how brief, from a book, journal, article or other written source without duly acknowledging the source
- copying any map, diagram, table or figure without duly acknowledging the source
- paraphrasing or otherwise using the ideas of another author without duly acknowledging the source
- re-using sections of verbatim text without using quote marks to indicate the text was copied from the source (even if a reference is given).

Other breaches of academic integrity that constitute cheating include but are not limited to:

- submitting work that is not a student's own, copying from another student, recycling another student's work, recycling previously submitted work, and working with another student in the same cohort in a manner that exceeds the boundaries of legitimate cooperation
- purchasing an assignment from a website and submitting it as original work
- requesting or paying someone else to write original work, such as an assignment, essay or computer program, and submitting it as original work.

Students who condone plagiarism and other breaches of academic integrity by allowing their work to be copied are also subject to student misconduct Rules.

Where proven, plagiarism and other breaches of misconduct are penalised in accordance with UTS Student Rules Section 16 – Student misconduct and appeals.

Avoiding plagiarism is one of the main reasons why the Faculty of Engineering and IT is insistent on the thorough and appropriate referencing of all written work. Students may seek assistance regarding appropriate referencing through UTS: HELPS.

Work submitted electronically may be subject to similarity detection software. Student work must be submitted in a format able to be assessed by the software (e.g. doc, pdf (text files), rtf, html).

Further information about avoiding plagiarism at UTS is available.

Retention of student work

The University reserves the right to retain the original or one copy of any work executed and/or submitted by a student as part of the course including, but not limited to, drawings, models, designs, plans and specifications, essays, programs, reports and theses, for any of the purposes designated in Student Rule 3.9.2. Such retention is not to affect any copyright or other intellectual property right that may exist in the student's work. Copies of student work may be retained for a period of up to five years for course accreditation purposes. Students are advised to contact their subject coordinator if they do not consent to the University retaining a copy of their work.

Statement on UTS email account

Email from the University to a student will only be sent to the student's UTS email address. Email sent from a student to the University must be sent from the student's UTS email address. University staff will not respond to email from any other email accounts for currently enrolled students.