SUBJECT OUTLINE

48024 Applications Programming

Course area	UTS: Engineering
Delivery	Autumn 2018; City
Credit points	6cp
Requisite(s)	48023 Programming Fundamentals OR 31267 Programming Fundamentals OR 31465 Object-oriented Programming
Result type	Grade and marks

Recommended studies: basic skills in Java programming

Subject coordinator

Dr Jaime Garcia

Email: Jaime.Garcia@uts.edu.au

Phone: 9514 4478

Location: CB11.07.127

Questions regarding assessment or content within the subject are welcome in lectures or tutorials or alternatively post them to the discussion board in UTSOnline. This helps ensure that all students get the benefit of the answers given.

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 email 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. Subject 32702, Request for Extension], and must be sent from your UTS email address.

Consultation hours: Check the UTSOnline 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 Jaime Garcia

Email: Jaime.Garcia@uts.edu.au

Details of other teaching staff will be published on UTSOnline under 'Staff Contacts'.

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. 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. Judge if a solution is well-designed
2. Design a good OO solution from a specification.
3. Demonstrate a working knowledge of lists in Java.

4. Use inheritance in Java.
5. Construct a GUI interface.

Course intended learning outcomes (CILOs)

This subject also contributes specifically to the development of the following faculty Course Intended Learning Outcomes (CILOs) and Engineers Australia (EA) Stage 1 competencies:

- Identify and apply relevant problem solving methodologies, which is linked to EA Stage 1 Competencies: 1.1, 2.1, 2.2, 2.3 (B.1)
- Design components, systems and/or processes to meet required specifications, which is linked to EA Stage 1 Competencies: 1.3, 1.6, 2.1, 2.2, 2.3 (B.2)
- Implement and test solutions, which is linked to EA Stage 1 Competencies: 2.2, 2.3 (B.5)
- Apply abstraction, mathematics and/or discipline fundamentals to analysis, design and operation, which is linked to EA Stage 1 Competencies: 1.1, 1.2, 2.1, 2.2 (C.1)
- Develop models using appropriate tools such as computer software, laboratory equipment and other devices, which is linked to EA Stage 1 Competencies: 2.2, 2.3, 2.4 (C.2)

Teaching and learning strategies

The subject has three contact hours a week in weeks 1, 5, 6, 10, 11 and 12, and two 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 seminars and the lab. Note that access to the lab material is electronically blocked until the student has completed the required weekly study module.

Regular face-to-face seminars present live programming demonstrations and examples of 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 Java event model

10. GUI Lists and list models

11. GUI tables and table models

Program

Week/Session	Dates	Description

1	12 March	**Topic**: Introduction to subject **Learning Material**: Watch all videos in the Study 1 module on PLATE. **Lab**: None; Labs will commence from week 2. **Activities to be completed this week**: 1. Watch the welcome video 2. Read the subject outline 3. Complete the self assessment Java quiz 4. Complete the 'Java Refresher' activities as per the quiz results
2	19 March	**Topic**: Basic patterns read, count, sum, read loop, output, max **Learning Material**: Watch all videos in the Study 2 module on PLATE. **Lab**: Working in small groups, devise pattern-based plans for a set of problems, then code them individually in Java. **Notes:** Study modules also must be completed before the labs, as PLATE will prevent you from accessing the lab exercises until after the study module has been completed.
3	26 March	**Topic**: Basic process key/framework incremental goals code, test and debug **Learning Material**: Complete the Study 3 module. **Lab**: Working in small groups, apply the basic processes to solve a set of problems, then code them individually in Java. **Notes:** Lab assessment 1
4	2 April	***** PUBLIC HOLIDAY **** **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**: Complete the Study 4 module. **Lab**: Working in small groups, break a large goal down into sub-goals, devise plans for each sub-goal, then code them individually in Java. **Notes:** Lab assessment 2

5	9 April	**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**: Complete the Study 5 module. **Lab**: A problem will be analysed in a tutor-lead discussion. Students will code the solution to the problem individually. **Notes:** Study modules should always be completed before the class so that you can understand the topical coding examples constructed during the class. Lab assessment 3 Assignment 1 released, worth 30%
6	16 April	**Topic**: Lists List concept List methods Patterns: lookup, match, menu **Learning Material**: Complete the Study 6 module. **Lab**: A list-based problem will be analysed in a tutor-lead discussion. Students will code the solution to the problem individually. **Notes:** Lab assessment 4
StuVac	23 April	
7	30 April	**Topic**: System design Syntax: Interfaces and superclasses Process review: putting it all together, location table **Learning Material**: Complete the Study 7 module. **Lab**: Your tutor can help with the assignment. **Notes:** The Study 7 module contains two kinds of content: 1. Videos which are to be watched before the lecture. 2. A development environment exercise which must be completed before the lab.

8	7 May	**Topic**: GUIs and events JavaFX stages, scenes, boxes, grid panes and GUI components Buttons, events and event handlers Patterns: the observer pattern **Learning Material**: Complete the Study 8 module. **Lab**: Build a set of small GUI programs. **Notes:** Assignment 1 due
9	14 May	**Topic**: The MVC architecture Model-View-Controller concept FXML, observable properties, property bindings Patterns: JavaFX property patterns **Learning Material**: Complete the Study 9 module. **Lab**: Your tutor will demonstrate with an example all of the steps required to build an MVC application, then you will code an MVC application following the same steps. **Notes:** Lab assessment 5 (*) Assignment 2 released, worth 20% (*) The techniques practiced and assessed in this lab will form the basis for Assignment 2, and so care should be taken to sufficiently prepare for it and to spend the required time to complete it.
10	21 May	**Topic**: GUI lists List views, observable lists, populating a list, list selection Controls Opening and closing windows Packages **Learning Material**: Complete the Study 10 module. **Lab**: You will extend your program from week 9 by adding GUI lists and multiple windows.
11	28 May	**Topic**: GUI tables Table views, populating a table, table selection Change listeners Exceptions **Learning Material**: Complete the Study 11 module.

		Lab: You will extend your program from week 10 by adding GUI Tables. Your tutor can also help with the assignment.
12	4 June	**Topic**: Review and sample exam **Lab**: Assignment demonstration and peer marking. Attendance is required. **Notes:** Assignment 2 due

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

Students are expected 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 assessment is individual. You can discuss approaches, and help with debugging, but you should write every line of code you submit except for subject code: lecture notes, reference solutions, lab solutions, and demo solutions. All assessments are submitted to the PLATE system at http://plate.it.uts.edu.au. Assessments that have a text-based user interface are marked by PLATE (Assignment 1 and Lab assessments 1-4). Assessments that have a graphical user interface may be marked by a combination of PLATE, student peer marking, and the subject coordinator (Assignment 2 and Lab assessment 5).

Assessment task 1: Labs

Intent: The labs, both assessed and non-assessed, provide practise 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):

B.2

Type: Laboratory/practical

Groupwork: Individual

Weight: 10%

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

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

Submit your solution to PLATE at https://plate.it.uts.edu.au.

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

Due: Not applicable

Criteria: All labs are marked for correctness only. The early labs are marked by PLATE, and the GUI lab by the Subject Co-Ordinator. The minimum essential requirement for a pass in any one lab is to pass more than half the assessed tasks. You do not have to pass the lab assessment to pass the subject.

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Functionality of design	25	1, 2, 3	B.2
Correctness of system modelling	25	1, 2, 3	B.2
Correctness of design	25	1, 2, 3	B.2
Functionality and correctness of design	25	1, 2, 3	B.2

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Further information: **Release date:** Each lab is released at the start of the scheduled lab class. Students must also complete the prerequisite study module on PLATE in order for the lab to be released.

Due date: Each assessed lab is due 167 hours after the release date. This means that a typical week's lab exercise is due 1 hour before the start of the following week's lab class. 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 labs will be accepted.

Assessment task 2: Programming assignment 1

Intent: This programming assignment provides practise 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):

B.1, B.2, B.5, C.1 and C.2

Type: Project

Groupwork: Individual

Weight: 30%

Task: The task is to develop a well-designed OO system that uses lists. The output of this task is a Java solution of one or more classes. You submit Java code for marking by PLATE at https://plate.it.uts.edu.au.

Due: 11.59pm Monday 7 May 2018

Criteria: The assignment is marked on correctness by PLATE, based on the match to the model solution’s IO trace.

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

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Application of methodology	0	1, 2, 3	B.1
Functionality of design	20	1, 2, 3	B.2
Correctness of system modelling	20	1, 2, 3	B.2
Correctness of design	20	1, 2, 3	B.2
Functionality and correctness of design	20	1, 2, 3	B.5
Correctness of application of theory	10	1, 2, 3	C.1
Logical design strategy	10	1, 2, 3	C.2

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Assessment task 3: Programming assignment 2

Intent: This programming assignment provides practise 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):

B.1, B.2, B.5, C.1 and C.2

Type: Project

Groupwork: Individual

Weight: 20%

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

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

You submit Java code for marking. This is marked by a combination of automated PLATE marking, peer marking and marking by the subject coordinator.

Submit your solution to PLATE at https://plate.it.uts.edu.au.

Due: 11.59pm Monday 4 June 2018

Criteria: The assignment is marked on correctness. The minimum essential requirement for a pass is to complete tasks worth more half the marks.

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Application of methodology	0	1, 2, 3, 4, 5	B.1
Functionality of design	20	1, 2, 3, 4, 5	B.2
Correctness of system modelling	20	1, 2, 3, 4, 5	B.2
Correctness of design	20	1, 2, 3, 4, 5	B.2

Functionality and correctness of design	20	1, 2, 3, 4, 5	B.5
Correctness of application of theory	10	1, 2, 3, 4, 5	C.1
Logical design strategy	10	1, 2, 3, 4, 5	C.2

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Assessment task 4: Final Examination

Intent:	The exam tests the factual knowledge taught in the subject.
Type:	Examination
Groupwork:	Individual
Weight:	40%
Task:	Answer 40 multiple choice questions.
Length:	Two hours
Criteria:	The exam is multiple choice, so the questions are marked for correctness only.

Moderation of marks

Marks provided to the student by PLATE will be moderated by the subject coordinator after the due date. Peer reviews by students will also be moderated by the subject coordinator after the peer review period has finished.

Assessment feedback

Lab assessments 1-4 and Assignment 1 are marked by PLATE, so you get feedback on the solution correctness as soon as you submit a solution; Lab assessment 5 and Assignment 2 use a GUI, so they cannot be marked by PLATE. 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.

Examination material or equipment

The examination is open book.

Minimum requirements

In order to pass this subject students must:

- Achieve a minimum of 50% in the final examination, and
- Achieve a minimum overall mark of 50%.

Students who do not meet these minimum requirements but achieve an overall mark of 50% or greater will fail the subject and receive their overall mark with an 'X' (fail) grade.

Prize offered

WiseTech Global Junior Programming Prize

References

You should study the weekly lecture videos and sample programs posted to PLATE.

Other resources

All subject material is posted in UTSOnline at https://online.uts.edu.au/

The following web sites 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.leepoint.net/notes-java/index.html ·
- http://www.bluej.org

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 http://tinyurl.com/upass2017 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 and EA Stage 1 competencies, refer to the FEIT Graduate Attributes webpage.

Assessment: faculty procedures and advice

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

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.

Querying marks/grades and Final Results

If a student disagrees with a mark or a final result awarded by a marker:

- where a student wishes to query a mark, the deadline for a query during teaching weeks is 10 working days from the date of the return of the task to the student
- where a student wishes to query a final examination result, the deadline is 10 working days from the official release of the final subject result.

Further information can be found at Academic advice.

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.

The ALO for undergraduate students is:

Chris Wong
telephone +61 2 9514 4501

The ALO for postgraduate students is:

Dr Nahm Tran
telephone +61 2 9514 4468

Statement about assessment procedures and advice

This subject outline must be read in conjunction with the policy and procedures for the assessment for coursework subjects.

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.