

## 41889 Application Development in the iOS Environment

**Course area** UTS: Information Technology

**Delivery** Autumn 2021; City

**Credit points** 6cp

**Requisite(s)** [48023](#) Programming Fundamentals OR [31267](#) Programming Fundamentals

**Result type** Grade and marks

Linked subjects: 42889 iOS Application Development

Attendance: Weeks 1-9: 1.5hpw (lecture), 1.5hpw (laboratory); Weeks 10-12: 1.5hpw (laboratory); Canvas Discussion Board.

Recommended studies: Students would benefit from completing other programming subjects, particularly 31251 Data Structures and Algorithms.

### Subject coordinator

Dr Hua Zuo

Email: [hua.zuo@uts.edu.au](mailto:hua.zuo@uts.edu.au)

Questions regarding assessment or content within the subject are welcome in lectures or tutorials or alternatively post them to the discussion board in Canvas. 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 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. Subject 41889, Request for Extension], and must be sent from your UTS email address.

### Teaching staff

Dr Hua Zuo

Email: [Hua.Zuo@uts.edu.au](mailto:Hua.Zuo@uts.edu.au)

### Subject description

This subject is an introduction to iOS application development. Students learn the Swift programming language, the iOS development environment and explore the iOS app frameworks. The subject has an emphasis on giving students practical exposure to software development in the iOS environment.

### Subject learning objectives (SLOs)

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

1. Design a well-structured programming solution that satisfies given software requirements.
2. Implement the solution using Swift programming language under the iOS development environment.
3. Independently explore and utilise various iOS frameworks and APIs.
4. Demonstrate the fully tested software using iOS simulator for iPads and iPhones.

### 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)
- Collaborative and Communicative: FEIT graduates work as an effective member or leader of diverse teams,

communicating effectively and operating within cross-disciplinary and cross-cultural contexts in the workplace. (E.1)

- Reflective: FEIT graduates critically self-review their performance to improve themselves, their teams, and the broader community and society. (F.1)

## Teaching and learning strategies

This subject is composed of formal lectures, video learning materials, practical programming exercises/projects, lab quizzes and a collaborative report.

The lectures will present new topics on iOS each week. Live code demonstrations will be provided in the lectures to enhance students' understanding of the topic. Students are encouraged to follow the demonstrations on their own computers whenever possible. Questioning by students is actively encouraged.

The video materials provide the detailed demonstration on how to use iOS development environment and focus on GUI programming. They will give student the opportunities to self study these topics prior to the lectures and the laboratory exercises.

The laboratories consist of small programming exercises that help student to get practical understanding of the software development concepts delivered in the lectures. The expected results from the exercises are provided so that students can check their progresses. There will be three small quizzes in the laboratories at various stages to check students' knowledge. This ensures students acquire necessary knowledge at a consistent pace. The last three laboratories will be dedicated to group collaboration work for Project 3. Laboratory assistants will be there to facilitate the sessions, answering questions, providing comment and encourage group collaboration.

The individual projects will present the students with the opportunity to practise the knowledge and the skills learned from the lectures and the laboratory exercises. The students will create small, useful, working programs independently. The projects will be assessed by both fellow students (peer assessment) and the laboratory assistants.

The group project ensures the students have the capabilities to collaboratively explore the vast amount of online iOS documentation and knowledge bases and apply the knowledge into practical usage. The project also encourages the students to learn from their peer group via peer assessment.

## Content (topics)

Topic 1: Swift

- Classes, inheritance, polymorphism
- Data type
- Loops and Conditionals

Topic 2: Cocoa Framework

- Foundation Framework
- Numbers, strings, collections
- File I/O
- GUI controls

Topic 3: iOS

- App lifecycle and design
- iOS frameworks
- Building and testing app for iOS simulator and device

## Program

Week/Session	Dates	Description
1	22 February	Introduction to iOS development  The Swift language and Xcode development environment  <b>Notes:</b>  Assessment Task 1 (individual project) releases

No tutorial

2	1 March	Introduction to Swift and Playground: Constants, Variables, Data Types, Operators, Control Flow
3	8 March	More Swift: Function, Loop, Structures, Classes, Collections Pseudo Code
4	15 March	More Swift: Optional, Type casting, Guard, Scope, Enumerations  <b>Notes:</b> Assessment Task 1 due Laboratory: Quiz 1
5	22 March	Introduction to UIKit framework Auto Layout
6	29 March	MVC Timer  <b>Notes:</b> Assessment Task 2 (individual project) releases
StuVac	5 April	Mid-session StuVac No lecture and lab
7	12 April	Navigation Controller Segues and Passing Data, Create UI Components Programmatically  <b>Notes:</b> Laboratory: Quiz 2
8	19 April	Protocols and TableView User defaults  <b>Notes:</b> Assessment Task 3 (group project) releases

9	26 April	Project planning Git Project management  <b>Notes:</b> Assessment Task 2 due
10	3 May	URL, Cocoapods  <b>Notes:</b> Laboratory: Quiz 3
11	10 May	Appstore CICD Testing
12	17 May	Group project presentation  <b>Notes:</b> Assessment Task 3 Due

## Assessment

### Laboratory Quiz

There will be three quizzes allocated at laboratory time during the semester.

### Project Submission

Projects will be submitted via Canvas by the due date.

### Project Demonstration

As part of marking your program, you will be required to demonstrate its functionality in your assigned laboratory. Details about this can be found on Canvas.

### Group Work

The first two assessment tasks are individual programs; the last project is a group report.

### Extensions

Extensions to the deadlines will be granted if there is a fully documented reason that merits it. The extension must be asked for before the due date.

## Assessment task 1: Individual Programming Project 1

<b>Intent:</b>	<p>The project will require the students to demonstrate their understanding and skills in programming including:</p> <ul style="list-style-type: none"> <li>• Data types</li> <li>• Loops and conditionals</li> <li>• Strings and arrays</li> <li>• Classes and OOP</li> <li>• Dynamic memory allocation and deletion</li> <li>• Testing and debugging</li> <li>• Using version control</li> </ul>
<b>Objective(s):</b>	<p>This assessment task addresses the following subject learning objectives (SLOs):</p> <p>1, 2 and 4</p> <p>This assessment task contributes to the development of the following Course Intended Learning Outcomes (CILOs):</p> <p>C.1</p>
<b>Type:</b>	Laboratory/practical
<b>Groupwork:</b>	Individual
<b>Weight:</b>	25%
<b>Task:</b>	<p>The project will require students to demonstrate their understanding and skills in programming including:</p> <ul style="list-style-type: none"> <li>• Data types</li> <li>• Loops and conditionals</li> <li>• Strings, arrays, and dictionaries</li> <li>• Testing and debugging</li> <li>• Using version control</li> </ul> <p>Program functionality will be evaluated by an automated testing suite, and code style will be reviewed manually.</p>
<b>Length:</b>	200-300 lines of code
<b>Due:</b>	<p>11.59pm Friday 19 March 2021</p> <p>Feedback for programming tasks (Assessment Tasks 1, 2 and 3) is generally given within 2 weeks of the due date.</p> <p>See also Further information.</p>
<b>Criteria:</b>	Details of the assessment criteria will be made available in the project specification.
<b>Further information:</b>	<p>PLEASE NOTE. The criteria weightings given in the Subject Outline are a rough approximation.</p> <p>Please see the Assessment Task Specification for a detailed explanation of how the criteria are combined.</p>

## Assessment task 2: Individual Programming Project 2

<b>Intent:</b>	<p>The project will require the students to demonstrate their understanding and skill in building a practical iPhone/iPad application solution from given customer requirements.</p> <ul style="list-style-type: none"> <li>• Designing the solution.</li> <li>• Building a GUI program.</li> <li>• Using the various GUI components.</li> <li>• Testing and debugging.</li> </ul>
<b>Objective(s):</b>	<p>This assessment task addresses the following subject learning objectives (SLOs):</p> <p>1, 2 and 4</p> <p>This assessment task contributes to the development of the following Course Intended Learning Outcomes (CILOs):</p> <p>C.1</p>
<b>Type:</b>	Laboratory/practical
<b>Groupwork:</b>	Individual
<b>Weight:</b>	30%
<b>Task:</b>	<p>The project will require students to demonstrate their understanding and skill in building a practical iPhone/iPad application, specifically:</p> <ul style="list-style-type: none"> <li>• Cocoa Touch UI frameworks</li> <li>• Apple Human Interface Guidelines</li> </ul> <p>Students will evaluate their peers' functionality, visual appeal, and "fun factor". Markers will rate usability and code style.</p>
<b>Length:</b>	300 – 400 lines of code
<b>Due:</b>	<p>11.59pm Friday 30 April 2021</p> <p>See also Further information.</p>
<b>Criteria:</b>	Details of the assessment criteria will be made available in the project specification.
<b>Further information:</b>	<p>PLEASE NOTE. The criteria weightings given in the Subject Outline are a rough approximation. Please see the Assessment Task Specification for a detailed explanation of how the criteria are combined.</p>

### Assessment task 3: Group Project

<b>Intent:</b>	This group project requires students to explore and understand specific iOS frameworks using available online resources, develop working code examples that utilise the iOS frameworks.
<b>Objective(s):</b>	<p>This assessment task addresses the following subject learning objectives (SLOs):</p> <p>2 and 3</p> <p>This assessment task contributes to the development of the following Course Intended Learning Outcomes (CILOs):</p> <p>C.1 and E.1</p>

<b>Type:</b>	Case study
<b>Groupwork:</b>	Group, group assessed
<b>Weight:</b>	30%
<b>Task:</b>	This group project requires students to explore and understand specific iOS frameworks using available online resources, develop working code examples, and explain how those frameworks solve specific problems. Student groups will describe their product in a final presentation.
<b>Length:</b>	About 5 min presentation.
<b>Due:</b>	11.59pm Friday 21 May 2021 See also Further information.
<b>Criteria:</b>	Details of the assessment criteria will be made available in the project specification.
<b>Further information:</b>	The project will be presented in your assigned laboratory in Week 12. PLEASE NOTE. The criteria weightings given in the Subject Outline are a rough approximation. Please see the Assessment Task Specification for a detailed explanation of how the criteria are combined.

#### **Assessment task 4: Quiz 1**

<b>Intent:</b>	Check students' knowledge on the subject and give feedback on progress.
<b>Objective(s):</b>	This assessment task contributes to the development of the following Course Intended Learning Outcomes (CILOs):  F.1
<b>Type:</b>	Quiz/test
<b>Groupwork:</b>	Individual
<b>Weight:</b>	5%
<b>Task:</b>	Online multiple choices questions.
<b>Length:</b>	20 to 30 minutes
<b>Due:</b>	In class in laboratory time, refer to the program.

#### **Assessment task 5: Quiz 2**

<b>Intent:</b>	Check students' knowledge on the subject and give feedback on progress.
<b>Objective(s):</b>	This assessment task contributes to the development of the following Course Intended Learning Outcomes (CILOs):  F.1

**Type:** Quiz/test

**Groupwork:** Individual

**Weight:** 5%

**Task:** Online multiple choices questions.

**Length:** 20 to 30 minutes

**Due:** In class in laboratory time, refer to the program.

### **Assessment task 6: Quiz 3**

**Intent:** Check students' knowledge on the subject and give feedback on progress.

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

F.1

**Type:** Quiz/test

**Groupwork:** Individual

**Weight:** 5%

**Task:** Online multiple choices questions.

**Length:** 20 to 30 minutes

**Due:** In class in laboratory time, refer to the program.

### **Assessment feedback**

Assessment feedback will be in two parts.

1. A submission system will be used that gives automated feedback about many features of the student's code. This is done with static analysis.
2. The marker will give individual feedback to the students.

Feedback will be given 2 weeks after assignment submission.

### **Minimum requirements**

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

### **Required texts**

There is no textbook. Most are out of date. Apart from lecture notes, students will be referred to iOS Developer Center website. See below. Students should familiarise themselves with the iOS Developer Centre, as well as common concepts of iOS development.

### **Other resources**

iOS Developer Center

<https://developer.apple.com/devcenter/ios/index.action>

There is a vast range of articles, downloads (including XCode), sample code, etc located here. Any iOS developer will



make extensive use of this resource.

Canvas

## 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 will be 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:

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