SUBJECT OUTLINE

48434 Embedded Software

Course area	UTS: Engineering
Delivery	Spring 2015; City
Subject classification	Field of practice: Electrical Engineering major
Credit points	6cp
Requisite(s)	48430 Embedded C
Result type	Grade and marks

Recommended studies: knowledge of the C language and digital systems is essential for this subject

Subject coordinator

Dr P McLean
Room: CB11.09.128
Phone: +61-2-9514-2339
Email: Peter.McLean@uts.edu.au

Teaching staff

Dr Ben Rodanski
Room: CB11.09.129
Phone: +61-2-9514-2426
Email: Ben.Rodanski@uts.edu.au

Subject description

This subject presents the theoertical and practical basis for the structure, operation and design of embedded software with an in-depth treatment of modern software design methodology. Software development involves some assembly language. The subject covers compiler and debugger tools; serial I/O and protocols; non-volatile memory; arithmetic operations; timing and interrupts; digital and analog interfacing; concurrent software; program optimisation; multi-module and multi-language programs; numerical techniques specific for certain tasks (such as the FFT and fuzzy logic control); real-time operating systems and Internet connectivity. The technical content is contextualised in a project in which students analyse the requirements of an embedded system and design the software to meet those requirements. Skills in debugging software are also developed through the practice-based nature of the subject.

Subject objectives

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

1. Design, write and test a variety of software modules found in modern embedded systems, such as: hardware abstraction layers; data structures; and interrupt service routines.
2. Design, write and test an embedded application that is modular, hierarchical, responsive to real-time requirements, and tightly constrained by time, size and cost.
3. Utilise a variety of software tools to write, execute and test embedded software applications.
4. Test software performance in an embedded system by selecting and using appropriate laboratory equipment.
5. Take responsibility for seeking out and evaluating knowledge from many sources.

This subject also contributes specifically to the development of the following course intended learning outcomes:

- Identify, interpret and analyse stakeholder needs [EA Stage 1 Competency: 1.2, 2.3, 2.4] (A.1)
- Identify and apply relevant problem solving methodologies [EA Stage 1 Competency:1.1, 2.1, 2.2, 2.3] (B.1)
- Design components, systems and/or processes to meet required specifications [EA Stage 1 Competency: 1.3, 1.6, 2.1, 2.2, 2.3] (B.2)
- Synthesise alternative/innovative solutions, concepts and procedures [EA Stage 1 Competency: 1.1, 3.3] (B.3)
- Apply decision making methodologies to evaluate solutions for efficiency, effectiveness and sustainability [EA Stage 1 Competency: 1.2, 2.1] (B.4)
- Implement and test solutions [EA Stage 1 Competency: 2.2, 2.3,] (B.5)
- Demonstrate research skills [EA Stage 1 Competency: 1.4, 2.1] (B.6)
- Apply abstraction, mathematics and/or discipline fundamentals to analysis, design and operation [EA Stage 1 Competency:1.1, 1.2, 2.1, 2.2] (C.1)
- Develop models using appropriate tools such as computer software, laboratory equipment and other devices [EA Stage 1 Competency: 2.2,2.3, 2.4] (C.2)
- Manage own time and processes effectively by prioritising competing demands to achieve personal goals [EA Stage 1 Competency: 3.5, 3.6] (D.1)
- Reflect on personal and professional experience to engage independent development beyond formal education for lifelong learning [EA Stage 1 Competency:3.3, 3.5] (D.2)
- Work as an effective member or leader of diverse teams within a multi-level, multi-disciplinary and multi-cultural setting [EA Stage 1 Competency:2.4, 3.2, 3.6] (E.2)
- Be able to conduct critical self-review and performance evaluation against appropriate criteria as a primary means of tracking personal development needs and achievements [EA Stage 1 Competency: 3.5] (F.1)
- Appreciate ethical implications of professional practice [EA Stage 1 Competency:3.1] (F.2)

Teaching and learning strategies

Class time is used for lectures, self-directed study sessions and laboratories. Lectures will introduce new material in a modular fashion that can then be applied to the design of a real embedded system. The laboratory work will be used to build up a complete software system in a step-by-step hierarchical manner. Towards the end of semester, students will undertake an individual project in which the class time will serve as valuable resource/design/discussion sessions.

There are three components to completing your study of Embedded Software.
They are:

- reading the lecture notes and associated “readings”
- attempting the laboratory assessment tasks
- completing the project satisfactorily

To guide you through these tasks there is a Timetable.

Learning in partnership
Using a fellow student as a learning partner has been found repeatedly to be an important learning support. The idea is that you contact a fellow student, by phone or whatever means is most convenient, to discuss your interpretation of a learning task, to check if your approaches are the same and to generally clear up any confusions which may have arisen. It has been found that well over half of the concerns students experience about their learning are to do with simply checking that they are 'on the right track' and can be solved using this method. If, however, the concern or uncertainty remains, it is then recommended that you contact your subject coordinator.

Your learning plan
Your time

Organising your time is a major challenge in learning. Leaving recommended readings and assignments to the last minute is a common problem. To assist you with this challenge you may find it useful to plan your study time before you start work on this subject. First decide on the best place and time each week to study without distractions and then make sure to adhere to your own plan.

It is estimated that over a period of 13 semester weeks you should set aside a total of approximately 8 hours of study time each week. It is recommended that you break up those hours into at least two study sessions each on a different day of the week. This is a rough guide only, as people learn at different rates and from different levels of experience.

Content

The technical content of the subject aims to develop the basic structure, operation and design of embedded software with an in-depth treatment of modern software design methodology. Software development will involve some assembly language. The subject covers compiler and debugger tools; serial I/O and protocols; non-volatile memory; arithmetic operations; timing and interrupts; digital and analog interfacing; concurrent software; program optimization; multi-module and multi-language programs; numerical techniques specific for certain tasks (such as the FFT and fuzzy logic control); real-time operating systems and Internet connectivity. The technical content is contextualised in a project in which students analyse the requirements of an embedded system and design the software to meet those requirements. Skills in debugging software will also be developed through the practice-based nature of the subject. Three engineering themes permeate the subject. The first theme is the need for a systems perspective in engineering – students will need to analyse and dissect (through a requirements specification) and eventually synthesise in a hierarchical manner (through software design). The second theme is related to the first in that students will be expected to draw knowledge from a wide variety of sources – previous subjects, industrial experience, new technology. The third theme is that of the need for engineers to take responsibility for their own professional development. Students will produce documentation to their software that conforms with a given standard. A practical examination at the end of the project also gives students experience in communicating technical ideas.

The content covered is divided into the following sections:

1. Prerequisite knowledge
2. Embedded Computer Systems
3. Software Development
4. Microcontroller Peripherals
5. Memory Types
6. Assembly Language
7. Interrupts
8. Concurrent Software
9. Timing Generation and Measurements
10. Digital and Analog Interfacing
11. Fixed-Point Processing
12. Embedded Control Systems

Each of these sections addresses an important aspect in modern embedded systems. The intention is that, as you work your way through the subject, your learning will be cumulative. That is, the content you cover in one section should directly help you to understand the topics that follow. A weekly learning schedule, based on a recommended study sequence of the sections, is given in the Study Guide. For each of the above sections, a separate list of topics and suggested reading is also provided in the Study Guide.

Below is a brief summary of the content that is later covered in detail in the lecture notes.

Section 1 – Prerequisite knowledge
You are expected to have successfully completed subjects in the C language and Introductory Digital Systems. This section lists the important topics that you are expected to have mastered in earlier subjects. Special attention is given to the application of the C language to microcontrollers with limited resources.

Section 2 – Embedded Computer Systems
The architecture of a popular 16-bit microcontroller is given, in terms of hardware modules and a programming model. Various features of the microcontroller are highlighted, including its memory map, serial communication interface, serial peripheral interface, enhanced capture timers, analog-to-digital converter, pulse width modulator, non-volatile EEPROM memory and Flash memory. Schematics for the embedded Modular Controller (ModCon) board will be given, showing various pieces of peripheral and interfacing hardware that will be used in the laboratory.

Section 3 – Software Development
Aspects of quality programming, self-documenting code, modular software development, layered software systems, device drivers, threads and debugging strategies will be covered. The operation of the C compiler and the assembler as well as the output files produced by both, will be examined.

Section 4 – Microcontroller Peripherals
Microcontrollers have a wealth of built-in peripherals. Some of these peripherals will be examined in depth. The clock and reset generator module will be examined in detail. Devices external to the microcontroller can communicate via serial ports or parallel ports. The serial communication interface (SCI) is often used to connect to external microcontrollers, modems or PCs. The serial peripheral interface (SPI) is used to connect to chips such as analog-to-digital converters, EEPROM and Flash memory, LED display drivers and many other special purpose chips. The encapsulation of external devices in software using “device drivers” will form part of the laboratory program.

Section 5 – Memory Types
Modern 16-bit microcontrollers have many types of memory, such as internal Flash, EEPROM and RAM; as well as special function registers that are memory-mapped peripherals. Utilising these different memories and registers requires special software techniques in both C and assembly language.

Section 6 – Assembly Language
Sometimes it is necessary to delve into assembly language, for reasons of efficiency, debugging, or executing code not supported by the C compiler. An overview of a complex instruction set computer (CISC) assembly language is given, with examples highlighting the various modes of memory addressing as well as the use of special instructions.

Section 7 – Interrupts
Interrupts are the key to building real-time embedded systems. The interrupt structure and hardware support for interrupts on a 16-bit microcontroller will be examined. Special compiler support for interrupt service routines will be highlighted.

Section 8 – Concurrent Software
Foreground and background threads will be covered, as well as multithreaded applications and the basis for real-time operating systems. The concept of shared resources and some mechanisms for accessing them using semaphores and critical sections will be discussed. The concept of thread scheduling will be reviewed.

Section 9 – Timing Generation and Measurements
The enhanced capture timer module of a 16-bit microcontroller will be examined to see how periodic interrupts are generated as well as how complicated external event capturing can be used to simplify software tasks. Frequency and period measurement are two important tasks of an embedded system, and techniques for both of them will be discussed.

Section 10 – Digital and Analog Interfacing
Standard parallel digital interfacing of external devices such as input switches and keyboards, liquid crystal displays and output LEDs is looked at in terms of hardware and software. A microcontroller often needs to handle analog data, such as an automatic control system, or a measurement system. The theory behind choosing a sample rate, data scaling and limiting, quantization and noise will be briefly examined. Methods for obtaining, operating on, and producing analog data at the required rate will be reviewed. Peripherals such as an analog-to-digital converter and a pulse width modulator (PWM) will be looked at in detail.

Section 11 – Fixed-Point Processing
Microcontrollers are limited in their data handling capabilities, as they often need to process data in real-time and do not possess hardware floating-point capabilities. Finite word length effects will be given and methods to overcome them will be examined. Some numerical methods will be presented for the integer evaluation of difficult results such as the square root.

Section 12 – Embedded Control Systems
Automatic control systems are a fact of modern life. This section looks at how a simple control system can be digitally implemented in an embedded system. The effect of system discretization, used in a digital implementation of a control system, is examined briefly. This forms part of the project background material.

Section 13 – Design Project
This section brings all the other sections together in a project that requires the analysis and design of an embedded system. You will be required to interpret specifications and come up with sound engineering designs using a variety of methods. The designs will be implemented and experimentally verified.

Program

Week/Session	Dates	Description
1A	27 Jul	***L1: Embedded Systems*** Overview of Embedded Systems. Overview of ModCon board. Freescale MC9S12A512 architecture.
1B	31 Jul	***L2: Embedded C*** Review of the C language. CodeWarrior IDE. Initializing and accessing I/O ports. Memory allocation. Self-documenting code. Modular software development. Layered software systems. Debugging.
2A	3 Aug	***L3: Microcontroller Peripherals*** Clocks and reset generator. Serial Communications Interface. PC USB Interface. FIFOs. Polling. ModCon serial protocol.
2B	7 Aug	
3A	10 Aug	***L4: Memory*** Flash memory. EEPROM. RAM. Special function registers. Memory-mapped peripherals.
3B	14 Aug	
4A	17 Aug	***L5: Interrupts*** Interrupts. Interrupt service routines. Hardware interrupts. Interrupt vectors and priority. Exceptions. Threads. Foreground and background threads. Re-entrant programming.
4B	21 Aug	Assessment: **Lab 1 Due**
5A	24 Aug	***L6: Timing Generation and Measurements*** Timer module. Modulus down-counter. Output compare. Input capture. Pulse accumulator.
5B	28 Aug	
6A	31 Aug	***L7: The Embedded Software Tool Chain*** Projects. Editing. Compiling. Libraries. Linking. The map file. Startup code. Vector table initialization. Loading. The S19 file.
6B	4 Sep	Assessment: **Lab 2 Due**

7A	7 Sep	***L8: Concurrent Software*** Threads. Schedulers. Operating systems. The semaphore. Mutual exclusion with semaphores. Synchronisation with semaphores. The producer / consumer problem with semaphores.
7B	11 Sep	
8A	14 Sep	***L9: Interfacing*** Input switches and keyboards. Analog to digital conversion. Digital to analog conversion. Serial Peripheral Interface.
8B	18 Sep	Assessment: **Lab 3 Due**
9A	21 Sep	***Tutorial Week***
9B	25 Sep	***Tutorial Week*** Assessment: **Mid-Semester Exam** Lectures L1-L8 inclusive.
VC	28 Sep	***Vice-Chancellor's Week***
VC	2 Oct	***Vice-Chancellor's Week***
10A	5 Oct	***Public Holiday***
10B	9 Oct	
11A	12 Oct	***Embedded Software Project*** Overview of project. [PROJECT HANDED OUT]
11B	16 Oct	Assessment: **Lab 4 Due**
12A	19 Oct	***L10: Fixed-Point Processing*** *Q*-notation. Other notations. Fixed-point calculations. Square-root algorithm for a fixed-point processor.
12B	23 Oct	

13A	26 Oct	***L11: Real-Time Operating Systems*** Real-time kernel concepts. Reentrancy. Thread priority. Mutual Exclusion. Synchronization. Interthread communication. Interrupts. Memory requirements. Advantages and disadvantages of real-time operating systems.
13B	30 Oct	Assessment: **Lab 5 Due**
14A	2 Nov	***Embedded Software Project*** Project work. [NO LECTURE]
14B	6 Nov	
15A	9 Nov	
15B	13 Nov	
16A	16 Nov	
16B	20 Nov	Assessment: **Project Due**

Additional information

Repeated Failure in this Subject

The Faculty takes repeated failures in a subject seriously and enforces Rule 10.6 of the University's Student and Related Rules:

http://www.gsu.uts.edu.au/rules/10-6.html

You should read these rules and be aware of the consequences of failure.

If you have failed **twice** before in this subject, then:

(i) You must seek advice from the Subject Coordinator. You will be asked to draw up and submit a study plan that outlines your strategy for passing this subject on the third attempt. A signed copy of this study plan will be kept by the Faculty for internal records.

(ii) If you do not seek advice from the Subject Coordinator by Week 2, then you do not have the Faculty's permission to enrol in the subject. If you stay enrolled in the subject then you will be breaking Rule 10.6.2 (1) of the University's Student and Related Rules.

(iii) You need to be aware that if you fail this subject for a third time, you will need to seek permission from the relevant Course Coordinator for any further enrolment in this subject (see below).

If you fail this subject for a **third** time, then:

(i) Subject Coordinators will deny permission for any further enrolment unless you can produce well-documented evidence that requires special consideration. In such cases, the Subject Coordinator will refer the matter to the relevant Course Coordinator, who will grant or deny enrolment for a fourth or subsequent attempt based on a student's overall performance in the course and the extent to which extenuating circumstances have contributed to one or more of the failures.

(ii) If you are granted permission for a fourth or subsequent attempt at this subject, then you must seek continuing assistance throughout this semester from the Subject Coordinator.

Assessment

Late Submission of Assessment Tasks

Late submission of an assessment task will attract a 10% penalty per day, up to a maximum of 10 days.

Assessment task 1: Lab 1

Intent: Skills in microcontroller modules, serial I/O, non-volatile memory, interrupt handling, analog interfacing and PC connectivity.

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

1, 2, 3 and 5

This assessment task contributes to the development of the following course intended learning outcomes:

A.1, B.1, B.2, B.3, B.4, B.5, B.6, C.1, D.1, D.2 and E.2

Type: Laboratory/practical

Groupwork: Group, group assessed

Weight: 8%

Task: Write a program that uses serial communication to transfer information between the Embedded Hardware and a PC

Due: Week 4
Session 4B

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Completeness of requirements specification	17	1, 2	A.1
Functionality of design	17	1, 2, 3	B.1, B.2, B.4, C.1
Correctness of application of theory	17	1, 2, 3	B.3
Correctness of design	17	1, 2, 3	B.5, B.6
Effectiveness of time management and independent learning	17	5	D.1, D.2
Efficiency of task performance	15	1, 2, 3	E.2

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Further information: Students will be assessed in a group of 2, and will be awarded the same mark.

Assessment task 2: Lab 2

Intent: Skills in microcontroller modules, serial I/O, non-volatile memory, interrupt handling, analog interfacing and PC connectivity.

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

1, 2, 3 and 5

This assessment task contributes to the development of the following course intended learning outcomes:

A.1, B.1, B.2, B.3, B.4, B.5, B.6, C.1, D.1, D.2 and E.2

Type: Laboratory/practical

Groupwork: Group, group assessed

Weight: 8%

Task: Write a program to utilise non-volatile-memory and system clocks

Due: Week 6
Session 6B

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Completeness of requirements specification	17	1, 2	A.1
Functionality of design	17	1, 2, 3	B.1, B.2, B.4, C.1
Correctness of application of theory	17	1, 2, 3	B.3
Correctness of design	17	1, 2, 3	B.5, B.6
Effectiveness of time management and independent learning	17	5	D.1, D.2
Efficiency of task performance	15	1, 2, 3	E.2

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Further information: Students will be assessed in a group of 2, and will be awarded the same mark.

Assessment task 3: Lab 3

Intent: Skills in microcontroller modules, serial I/O, non-volatile memory, interrupt handling, analog interfacing and PC connectivity.

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

1, 2, 3 and 5

This assessment task contributes to the development of the following course intended learning outcomes:

A.1, B.1, B.2, B.3, B.4, B.5, B.6, C.1, D.1, D.2, E.2, F.1 and F.2

Type: Laboratory/practical

Groupwork: Group, group assessed

Weight: 8%

Task: Write a program that uses interrupts and timers

Due: Week 8
Session 8A

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Completeness of requirements specification	14	1, 2	A.1
Functionality of design	14	1, 2, 3	B.1, B.2, B.4, C.1
Correctness of application of theory	14	1, 2, 3	B.3
Correctness of design	14	1, 2, 3	B.5, B.6
Effectiveness of time management and independent learning	14	5	D.1, D.2
Efficiency of task performance	14	1, 2, 3	E.2
Evidence of benchmarking	16	1, 2, 3	F.1, F.2

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Further information: Students will be assessed in a group of 2, and will be awarded the same mark.

Assessment task 4: Lab 4

Intent: Skills in microcontroller modules, serial I/O, non-volatile memory, interrupt handling, analog interfacing and PC connectivity.

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

1, 2, 3 and 5

This assessment task contributes to the development of the following course intended learning outcomes:

A.1, B.1, B.2, B.3, B.4, B.5, B.6, C.1, D.1, D.2, E.2, F.1 and F.2

Type: Laboratory/practical

Groupwork: Group, group assessed

Weight: 8%

Task: Write a program that uses serial protocols and other digital interfaces

Due: Week 11
Session 11B

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Completeness of requirements specification	14	1, 2	A.1
Functionality of design	14	1, 2, 3	B.1, B.2, B.4, C.1
Correctness of application of theory	14	1, 2, 3	B.3
Correctness of design	14	1, 2, 3	B.5, B.6
Effectiveness of time management and independent learning	14	5	D.1, D.2
Efficiency of task performance	14	1, 2, 3	E.2
Evidence of benchmarking	16	1, 2, 3	., F.1, F.2

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Further information: Students will be assessed in a group of 2, and will be awarded the same mark.

Assessment task 5: Lab 5

Intent: Skills in microcontroller modules, serial I/O, non-volatile memory, interrupt handling, analog interfacing and PC connectivity.

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

1, 2, 3 and 5

This assessment task contributes to the development of the following course intended learning outcomes:

A.1, B.1, B.2, B.3, B.4, B.5, B.6, C.1, D.1, D.2 and E.2

Type: Laboratory/practical

Groupwork: Group, individually assessed

Weight: 8%

Task: Write a program that uses a human-machine interface

Due: Week 13
Session 13B

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Completeness of requirements specification	17	1, 2	A.1
Functionality of design	17	1, 2, 3	B.1, B.2, B.4, C.1
Correctness of application of theory	17	1, 2, 3	B.3
Correctness of design	17	1, 2, 3	B.5, B.6
Effectiveness of time management and independent learning	17	5	D.1, D.2
Efficiency of task performance	15	1, 2, 3	E.2

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Further information: Students will be assessed in a group of 2, and will be awarded the same mark.

Assessment task 6: Mid-semester exam

Intent: Test knowledge of C in an embedded environment as well as knowledge of a particular microcontroller's peripherals.

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

1 and 2

This assessment task contributes to the development of the following course intended learning outcomes:

B.1, B.2, B.3 and C.1

Type: Mid-semester examination

Groupwork: Individual

Weight: 20%

Task: Mid-semester exam

Due: Week 8
Session 8B

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Functionality of design	33	1, 2	B.1, B.2
Correctness of application of theory	33	1, 2	B.3
Correctness of application	34	1, 2	C.1

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Further information: This task will be assessed on an individual basis.

Assessment task 7: Project

Intent: To analyse a set of specifications, design, simulate, test and practically demonstrate a simple embedded system. Skill in maintaining a record of an engineering design, reflecting on laboratory experiences, recording test results, and documenting software designs.

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

1, 2, 3, 4 and 5

This assessment task contributes to the development of the following course intended learning outcomes:

A.1, B.1, B.2, B.3, B.4, B.5, B.6, C.1, C.2, D.1 and D.2

Type: Laboratory/practical

Groupwork: Individual

Weight: 40%

Task: Write a program that implements a simple application specific embedded system.

Due: Week 16
Session 16B

Criteria: In assessing your performance in this subject, we will be looking for evidence that:

- You are able to efficiently carry out an accurate analysis of the requirements of embedded systems which are similar to those dealt with in this subject.
- You have understood the concepts used in embedded software design and are able to apply them to the design of practical systems such as simple embedded systems.
- You can distinguish between the different methods of implementing a real-time embedded system and know their limitations and how to apply them correctly.
- You have understood the methods of hardware interfacing and are able to apply them to practical embedded systems such as those covered in this subject.

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Completeness of requirements specification	17	1, 2	A.1
Functionality of design	17	1, 2, 3	B.1, B.2, B.4, C.1
Correctness of application of theory	17	1, 2, 3	B.3
Correctness of design	17	1, 2, 3, 4	B.5, B.6
Effectiveness of time management and independent learning	17	5	D.1, D.2
Relevance of model	15	1, 2	C.2

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Further information: This task will be assessed on an individual basis.

Assessment feedback

Labs: individual detailed feedback, formative and summative

Mid-Semester Test: returned work, summative with feedback

Project: returned work, summative

Graduate attribute development

For full details about the Faculty's graduate attributes, see:

www.uts.edu.au/about/faculty-engineering-and-information-technology/who-we-are/welcome/what-makes-us-unique-0

Assessment: faculty procedures and advice

Special consideration

Information on special consideration and special needs can be found at:

www.uts.edu.au/current-students/managing-your-course/classes-and-assessment/special-circumstances

Special consideration requests are submitted and resolved through the UTS Special Consideration Process.

Special needs

Students should email the subject coordinator as soon as possible (and prior to the assessment deadline) to indicate how their ability to meet an assessment component or requirement is impacted, and that they are seeking assistance through UTS Special Needs as detailed in Section 5.1.3 of Procedures for the Assessment of Coursework Subjects.

Academic integrity

Students should refer to the Advice to Students on Good Academic Practice policy at:

www.gsu.uts.edu.au/policies/academicpractice.html

If your tutor, assessor or lecturer suspects that you have plagiarised and/or cheated in any assessment task, they have no choice under UTS rule 16.6 but to refer the matter to the Responsible Academic Officer (usually the Director of Undergraduate or Postgraduate Programs, or the Associate Dean Teaching and Learning).

Academic liaison officer

Academic Liaison Officers (ALOs) are academic staff in each faculty who assist three groups of students: students with disabilities or ongoing illness; students who have difficulties in their studies because of their family commitments (e.g. being a primary carer for small children or a family member with a disability); and students who gained entry through the UTS Educational Access Scheme or Special Admissions.

ALOs are responsible for determining alternative assessment arrangements for students with disabilities. Students who are requesting adjustments to assessment arrangements because of their disability or illness are requested to see a Disability Services Officer in the Special Needs Service before they see their ALO.

The ALO for Engineering students is:
Dr Bruce Moulton
telephone +61 2 9514 2681
email Bruce.Moulton@uts.edu.au

The ALO for IT students is:
Dr Bruce Moulton
telephone +61 2 9514 2681
email Bruce.Moulton@uts.edu.au

Support

Improve your academic and English language skills: HELPS (Higher Education Language and Presentation Support) Service provides assistance with English language proficiency and academic language. Students who need to develop their written and/or spoken English should make use of the free services offered by HELPS, including academic language workshops, vacation intensive courses, drop-in consultations, individual appointments and Conversations@UTS (www.uts.edu.au/current-students/support/helps/about-helps).

HELPS is located in Student Services, on level 3 building 1. Phone 9514 9733.

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, available at:

www.gsu.uts.edu.au/policies/assessment-coursework.html

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 an examination result, the deadline is 10 working days from the official release of the final subject result.

More information can be found at:

https://my.feit.uts.edu.au/pages/course/student_policies_rules

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 Rule 3.9.2 (www.gsu.uts.edu.au/rules/student/section-3.html#r3.9). Such retention is not to affect any copyright or other intellectual property right that may exist in such student 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.

Disclaimer

This outline serves as a supplement to the Faculty of Engineering and Information Technology Student Guide. On all matters not specifically covered in this outline, the requirements specified in the Student Guide apply.