SUBJECT OUTLINE

31927 Application Development with .NET

Course area	UTS: Information Technology
Delivery	Spring 2022; City
Credit points	6cp
Requisite(s)	48024 Applications Programming
Result type	Grade and marks

Attendance: 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: a good understanding of programming constructs, O-O concepts and programming practices

Subject coordinator

Dr Avinash Singh (Lecturer)
Room: CB11.07.108
Ph: (02) 9514 4426
Email: avinash.singh@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 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 on Canvas
Step 2: Ask peers in your group or other groups or via 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. Subject 31927, Request for Assignment 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

Lecturer:
Dr. Avinash Singh
Email: avinash.singh@uts.edu.au

Tutors:
Information available on Canvas.

Subject description

This subject introduces C#, Visual Studio and the .NET development environment. The emphasis is on examining the .NET framework and the practicalities of developing software in this setting using the C# language.

Subject learning objectives (SLOs)

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

1. Understand the .NET framework architecture.

2. Understand how the .NET framework implements OO concepts and the implications this has for new language design.
3. Understand the mechanisms and techniques for the deployment and configuration of .NET applications.
4. Be able to utilize the .NET libraries.
5. Design and develop a C# program in the .NET environment.

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

1.5 hour lecture
1.5 hour lab
UTS Online Discussion board

Content (topics)

1. The .NET framework Common Language Runtime, Common Type System, Common Language Specification
2. Introduction to C# within the .NET Framework
3. Programming C# .NET
4. Windows Forms and Controls
5. Object Oriented Programming and Design concepts in the .NET Framework
6. Debugging techniques in the .Net Framework
7. Data structures in the C# .NET
8. LINQ to Objects

Program

Week/Session	Dates	Description
1	01 Aug	**Lecture 1:** Introduction to Application Development with .NET' and .NET Overview Preparation Activities: Students will be introduced to the subject and will have clear idea of the following: • what to expect in the following sessions • how the lab tutorials and quizzes looks like • what to prepare before the lectures/labs/tutorials • the assignments tasks, their deadlines and marking guidelines • how to seek help when encountering problems in learning the subject

2	08 Aug	**Lecture 2:** Programming in C# 1 Students will learn about programming basics in C#: such as data types, variable & constants, value/reference type, input/output, operators, conditions & loops, strings and Enums, with examples and lab tasks. **Notes:** **Quiz 1:** In Lab/Tutorial session.
3	15 Aug	**Lecture 3:** Programming in C# 2 A brief recap of the different programming paradigms will be the provided. Students will explore how to use arrays in C#, creating user defined methods, Object-oriented programming basics with respect to C#, creating classes/objects, methods and constructors.
4	22 Aug	**Lecture 4:** Programming in C# 3 Students will explore further topics on Object Oriented programming in C# (Polymorphism, method/constructor/operator overloading, etc), accessors, garbage collection, destructor, Structs, file handling, etc. **Notes:** **Quiz 2:** In Lab/Tutorial session.
5	29 Aug	**Lecture 5:** Programming with C# 4 Topics such as Testing, Debugging, Exception handling, Arraylist etc. will be introduced with examples and tasks in lab.
6	05 Sept	**Lecture 6:** Programming in C# 5 Topics such as Inheritance, method overriding/hiding, abstract/sealed classes & methods, and Interfaces will be discussed in detail. **Notes:** **Quiz 3:** In Lab/Tutorial session.
7	12 Sept	**Lecture 7:** Windows Forms -1 Students will learn how to create Windows Forms and add different controls on the GUI. Event handling basics, different controls will be introduced with examples and tasks in labs
8	19 Sept	**Lecture 8:** Windows Forms -2 Use of MessageBox, Dialogs, multiple form handling, will the discussed with examples. Introduction to Generics will be provided. **Notes:** **Quiz 4:** In Lab/Tutorial session.

		Assignment-1 due on Monday 19 Sept 2022, 11.00am, this week. Code to be submitted via Canvas by the due date. Functionality to be demonstrated in labs.
Stuvac	26 Sept	**Lecture 9:** Collections and Generics **Notes:** **Stuvac:** All 3rd Oct classes/labs are *moved to Stuvac due to public holiday on Oct. 3, 2022, however labs on Tuesday will next week as planned (no change).* *No Classes/Labs on Oct. 3, 2022.*
9	03 Oct	**Public Holiday on Monday, 03 Oct 2022: No class\lab on Monday, however, labs on Tue will be as planned** **Notes:** *No lab or class on Monday.* *Labs on Tue will continue as planned this week.*
10	10 Oct	**Lecture10:** Delegates, Anonymous methods, Lambda and LINQ
11	17 Oct	**Lecture11:** .Net in Industry: Special/Guest Lecture **Notes:** **Quiz 5:** In Lab/Tutorial session.
12	24 Oct	**Lecture 12:** Special Lecture Students will be introduced to database access using C#, etc. **Notes:** **Assignment 2 due on Monday, 24 Oct. 2022, 11.00am, this week. Code to be submitted via Canvas by the due date.** **Functionality to be demonstrated in labs this week.**

Assessment

Assignment Submission and Return:
Assignments will be submitted via UTSOnline by the due date and be ready for demonstration in the lab on the specified week. Full details will be given in the assignment specification.

Assignment Feedback:
Consistent with UTS policy, students will receive feedback in a timely manner that assists them to understand the learning objectives achieved and how they could improve the quality of their work. Feedback on assessments will given by tutors in labs. Additionally, the final marks will also include feedback from the markers.

Group Work:
Both assignments are individually developed and marked. There is NO group work.

Missing Assessment:

If you miss any piece of assessment through documented illness or misadventure, you should consult with the Subject Coordinator. In the case of the final examination, your "Application for Alternative Examination (due to Illness or Misadventure)" form must be submitted with appropriate documentation to the Student Information & Administration Centre within three working days of the examination.

Assessment task 1: Programming Assignment

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

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: Exercises

Groupwork: Individual

Weight: 35%

Task: Develop a Visual Studio console application in C#.

Once completed, the entire Visual Studio project will be submitted via UTS Canvas.

The assignment will then be formally demonstrated in the students assigned laboratory the week after the due date.

A complete description of the assignment will be given on the release date.

Length: 300 or more lines of code. The exact number will depend on how the students solves the task.

Due: 11.00am Monday 19 September 2022 Code to be uploaded to UTS Canvas by 11.00 am on the due date. Demonstrations: During the Labs of Week 8 and 9. See also Further information.
See also Further information.

Further information: PLEASE NOTE. The criteria weightings given in the Subject Outline are a rough approximation. Please see the Assignment Specification for a detailed explanation of how the criteria are combined

Assessment task 2: Programming Assignment

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

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: Exercises

Groupwork: Individual

Weight: 35%

Task: Develop a Visual Studio Windows form application in C#.

Once completed, the entire Visual Studio project will be submitted via UTS Canvas.

The assignment will then be formally demonstrated in the students assigned laboratory the week after the due date.

A complete description of the assignment will be given on the release date.

Length: 300 or more lines of code. The number of lines will depend on the solution the student develops for the assignment.

Due: 11.00am Monday 24 October 2022 Code to be uploaded to UTS Canvas by 11.00 am on the due date. Demonstrations: During the Labs of Week 12. See also Further information.
See also Further information.

Further information: PLEASE NOTE. The criteria weightings given in the Subject Outline are a rough approximation. Please see the Assignment Specification for a detailed explanation of how the criteria are combined

Assessment task 3: Quizzes

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

1 and 2

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

D.1

Type: Quiz/test

Groupwork: Individual

Weight: 30%

Task: There will be multiple-choice questions. Calculators are not permitted.

Due: In class in Tutorial/lab session. Check program for details. It is a closed book quiz, use of text books, class notes/slides, online materials are not permitted.

Moderation of marks

Programing assessments are marked against objective criteria and so moderation is not needed. For the exam, free text answers will be moderated by a member of the teaching staff to ensure consistency in marking.

Minimum requirements

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

Recommended texts

Links to online resources will be provided on UTSOnline.

Other resources

A UTSOnline workspace will be available to enrolled students providing tools to support collaboration and communication with staff and other students. UTSOnline can be found at: online.uts.edu.au.

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:

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.