Unit Outline 2012
Faculty of Information Sciences and Engineering

Software Technology 1 G (replacing Computers and Programming 6686)

8995
This Unit Outline must be read in conjunction with:

a) *UC Student Guide to Policies*, which sets out University-wide policies and procedures, including information on matters such as plagiarism, grade descriptors, moderation, feedback and deferred exams, and is available at *(scroll to bottom of page)*

b) *UC Guide to Student Services*, and is available at *(scroll to bottom of page)*

c) Any additional information specified in section 6h.

---

**1: General Information**

1a Unit title: Software Technology 1G

1b Unit number: 8995

1c Semester and year offered: S2, 2012

1d Credit point value: 3 CP

1e Unit level: Level 1 UG

1f Name of Unit Convener and contact details (including telephone and email)

   Dr. Girija Chetty
   Room no. 11C48
   Ph: 6201 2512
   girija.chetty@canberra.edu.au

   Unit Moderator and contact details (including telephone and email)

   Prof. Michael Wagner
   Room no. 11C35
   Ph: 6201 5477
   michael.wagner@canberra.edu.au

1g Administrative contact details (including name, location, telephone and email)

   Faculty Office
   Room no. 11B14
   School of ISE,
   Ph: 6201 2619, 6201 2417
   ise@canberra.edu.au
2: Academic Content

2a Unit description and learning outcomes

The unit provides a contextual approach to the art and science of software construction, introducing sufficient theory and practice of computer programming to enable further studies in computing. It will stress fundamental issues in the design and construction of programs in general. Topics covered include: algorithm design; problem solving within an application framework; classes / objects; object interaction; modular program construction including the principles of problem oriented data abstraction; encapsulation and information hiding; inheritance and polymorphism; file input and output; reusable component libraries; program testing; with emphasis on the production of simple, reliable and maintainable software.

Learning outcomes

On successfully completing the unit, a student will be able to:

1. Learn the principles of object oriented programming and apply it to build a complete and correct application.

2. Including advanced concepts in object oriented programming such as encapsulation and information hiding; inheritance and polymorphism for developing reusable software components.

3. Learn the basic concepts of testing and validation in the context of object oriented software system.

4. Be able to apply basic object orient design patterns such as model view controller patterns for the development of object oriented software components.

2b Generic skills

Graduates are expected to be able to:

1. Identify problems and analyse the main features of problems relevant to their professional field;

2. Apply appropriate problem solving processes, arguments, critical and creative thinking;

3. Implement and evaluate strategies for the resolution of problems;

4. Anticipate and define new problems; and

5. Identify and resolve new problems in new fields.
2c Prerequisites and/or co-requisites

(a) Introduction to Information Technology (4478) Or

(b) Introduction to Software Technology Or

Appropriate skills in programming

These may have been achieved by studying an appropriate introductory programming course in the previous years of study or work experience. If you have any questions about your prerequisites, you should consult with the Unit Convener. (Dr. Girija Chetty)

3: Delivery of Unit and Timetable

3a Delivery mode

The unit will be delivered face-to-face on campus with two 1.5-hour lectures and one 2-hour tutorial / laboratory per 2 weeks (fortnight).

3b Timetable of activities, such as lectures/ tutorials/ practicals/ field classes, showing key dates and topics

<table>
<thead>
<tr>
<th>Week</th>
<th>Activity</th>
<th>Lecture /Tutorial/Laboratory</th>
<th>Tutorial</th>
<th>Assessment Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to Computers and Programming (Chapter 1)</td>
<td>No Tutorial</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Introduction to Visual C# (Chapter 2)</td>
<td>Tutorial 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Processing Data (Chapter 3)</td>
<td>Tutorial 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Making Decisions (Chapter 4)</td>
<td>Tutorial 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Loops, Files &amp; Random Numbers (Chapter 5)</td>
<td>Tutorial 4</td>
<td>Class Test 1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Modularizing your Code with Methods (Chapter 6)</td>
<td>Tutorial 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Arrays and Lists (Chapter 7)</td>
<td>Tutorial 6</td>
<td>Assignment 1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Mid Semester Break</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>More about processing data (Chapter 8)</td>
<td>Tutorial 7</td>
<td>Class Test 2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Classes and Multiform Projects (Chapter 9)</td>
<td>Tutorial 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Inheritance and Polymorphism (Chapter 10)</td>
<td>Tutorial 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Databases (Chapter 11)</td>
<td>Tutorial 10</td>
<td>Assignment 2</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Revision</td>
<td>Tutorial 11</td>
<td>Class Test 3</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Sample Exam Discussion</td>
<td>Tutorial 12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4: Unit Resources

4a Lists of required texts/readings

Text Book:

Starting out with Visual C# 2010, 2/E
Tony Gaddis,
ISBN-10: 0132165457
Publisher: Addison-Wesley
Copyright: 2012

Location: UCU Bookshop

Additional References:

- C# Software Solutions: Foundations of Program Design
  John Lewis, Villanova University
  ISBN-10: 0321267168
  Publisher: Addison-Wesley
  Copyright: 2007

- Modern software development using C# .NET
  Author Wiener, Richard,
  Published Boston, MA : Brooks/Cole : Thomson Learning, c2007

- C# : how to program / H.M. Deitel ... [et al.].
  Publisher Prentice Hall, Upper Saddle River, N.J.

4b Materials and equipment

Software: The software used in the laboratories for working on different assessment
tasks in this unit will be the C# programming language with Visual Studio.NET 2010
Integrated Development Environments. University has a site licence for this software,
hence it is available in computer labs in the University and can be loaned from
Faculty office for students enrolled in the unit.

4c Unit website

The unit will be administered through UC online learning environment Moodle, which
can be accessed at http://learnonline.canberra.edu.au/
5: Assessment

5a Assessment overview

<table>
<thead>
<tr>
<th>Assessment Item (including exams held in the exam period)</th>
<th>Due Date of Assignments</th>
<th>Weighting (total to equal 100%)</th>
<th>Addresses learning outcome(s)</th>
<th>Addresses generic skill(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment 1 – C# Programming</td>
<td>5 pm Monday Week 7</td>
<td>15 %</td>
<td>1 &amp; 3</td>
<td>1,2,3</td>
</tr>
<tr>
<td>Assignment 2 – C# Object Oriented Programming</td>
<td>5 pm Monday Week 12</td>
<td>15 %</td>
<td>2, 3 &amp; 4</td>
<td>3,4 &amp; 5</td>
</tr>
<tr>
<td>Class Test 1</td>
<td>In Class - Week 5</td>
<td>10 %</td>
<td>1 &amp; 3</td>
<td>1,2,3</td>
</tr>
<tr>
<td>Class Test 2</td>
<td>In Class - Week 9</td>
<td>10 %</td>
<td>2, 3 &amp; 4</td>
<td>3,4 &amp; 5</td>
</tr>
<tr>
<td>Class Test 2</td>
<td>In Class - Week 13</td>
<td>10 %</td>
<td>2, 3 &amp; 4</td>
<td>3,4 &amp; 5</td>
</tr>
<tr>
<td>Final Examination</td>
<td>Exam period</td>
<td>40 %</td>
<td>1,2,3 &amp; 4</td>
<td>1,2,3,4,5</td>
</tr>
</tbody>
</table>

5b Details of each assessment item

- Examination: 3 hour examination; two A4 sides of handwritten notes permitted.
- Each class test will be held in the tutorial/Lab class. There are no notes allowed.
- The details of each assignment will be given on each assignment handout. Late Assignments (without prior approval from the lecturer) will receive a penalty of 10% of the value of the assignment for each day late. An assignment submitted after one week of the assignment due date will attract a maximum of 50%.
- Assignments are meant to be individual work, although talking a problem over with another student or tutor is considered one reasonable way of learning. However, the actual implementation must be your own work. Assignments which are found to be duplicates will have the marks divided among them. Experience has shown that students who do not do their own work are unlikely to pass the exam (and therefore the unit).
- It is highly recommended that students participate in all activities including class room lectures, tutorials and labs. There will be marks for participation and demonstration of lab activities and assignments as part of Assignment 1 and 2 marks.
- It is recommended that students who did not submit any internal assessment activities or participate in the unit throughout the semester do not apply for deferred exam.

5c Special assessment requirements
1. In order to pass the unit, students need to sit for all tests and the exam, and have no outstanding resubmissions by the exam time.

2. In the case of illness, misadventure or unavoidable commitments at the scheduled time of an examination, students should contact the Unit Convener as soon as possible. (well in advance).

3. Grades: For final assessment in the subject, the result will be one of the following grades: HD, DI, CR, P or Fail.

4. Performing satisfactorily in the assignments means there are no outstanding submissions by the exam time.

Subject to meeting the above requirements, your grade will depend on your exam mark (EM) and the assignments mark (AM), where
- Exams mark (EM) = Final Exam Mark converted to percentage.
- Assignment mark (AM) = (A1 + A2 + Class Test1 + Class Test2 + Class Test 3 mark) converted to percentage.

Grades will be awarded according to the following table:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage Marks in each of the Exams (EM) Assignment (AM) components</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD</td>
<td>EM in the range ≥ 85 &amp; AM ≥ 85</td>
</tr>
<tr>
<td>DI</td>
<td>EM in the range ≥ 75 &amp; AM ≥ 75</td>
</tr>
<tr>
<td>CR</td>
<td>EM in the range ≥ 65 &amp; AM ≥ 65</td>
</tr>
<tr>
<td>P</td>
<td>EM in the range ≥ 50 &amp; AM ≥ 50</td>
</tr>
<tr>
<td>NX</td>
<td>EM in the range &lt; 50 OR AM &lt; 50</td>
</tr>
</tbody>
</table>

5d Supplementary assessment

To be eligible to undertake supplementary assessment in a unit, a student must:
- be enrolled in their final semester of study;
- have failed a single unit, with a final mark between 45-49% in the unit;
- have passed all other units undertaken in that semester.

The failed unit must be the final unit required to complete the academic requirements of their course.

5e Academic Integrity

Students should uphold University standards on ethical scholarship. Good scholarship involves building on the work of others and use of others work must be acknowledged with proper attribution made. Cheating, plagiarism, and falsification of data are dishonest practices which contravene academic values.
6: **Student Responsibility**

6a **Workload**
The amount of time you will need to spend on study in this unit will depend on a number of factors including your prior knowledge, learning skill level and learning style. Nevertheless, in planning your time commitments you should note that for a 3cp unit the total notional workload over the semester or term is assumed to be 150 hours. These hours include time spent in classes. The total workload for units of different credit point value should vary proportionally. For example, for a 6cp unit the total notional workload over a semester or term is assumed to be 300 hours.

The student workload for this unit is:

- a. Lectures : \(13 \times 2 \times 1.5h = 39h\)
- b. Tutorials/Laboratories : \(6 \times 2 = 12h\)
- c. Homework \(7 \times 2 = 14h\)
- d. Preparation (Lectures/Tutorials/Labs) \(13 \times 2 = 26h\)
- e. Assignment 1 = 10h
- f. Assignment 2 = 10h
- g. Class Test 1 = 6h
- h. Class Test 2 = 6h
- i. Class Test 3 = 6h
- j. Final Exam = 21h

Total 150 hours

6b **Special needs**

Students who need assistance in undertaking the unit because of disability or other circumstances should inform their Unit Convener or UC AccessAbility (formerly the Disabilities Office) as soon as possible so the necessary arrangements can be made.

6c **Attendance requirements**

Experience has shown that students who do not attend the lectures and tutorials will have difficulty in passing the subject.

6d **Withdrawal**
If you are planning to withdraw please discuss with your unit convener. Please see this link for further information on deadlines.

6e **Required IT skills**

Please refer to section 2b.
6f Costs

Text book and consumables

6g Work Integrated Learning

6h Additional information

It is important that students refer to Unit Website (through Moodle – UC online learning environment) on a regular basis for any variations in the schedule and deadlines for the assessment tasks.

7: Student Feedback

All students enrolled in this unit will have an opportunity to provide anonymous feedback on the unit at the end of the Semester via the Unit Satisfaction Survey (USS) which will be presented to you on OSIS. Your lecturer or tutor may also invite you to provide more detailed feedback on their teaching through an anonymous in-class questionnaire administered through the University’s Teaching and Learning Centre (TLC).

8: Authority of this Unit Outline

Any change to the information contained in Section 2 (Academic content), and Section 5 (Assessment) of this document, will only be made by the Unit Convener if the written agreement of Head of Discipline and a majority of students has been obtained; and if written advice of the change is then forwarded to each student enrolled in the unit at their registered term address. Any individual student who believes him / herself to be disadvantaged by a change is encouraged to discuss the matter with the Unit Convener.