Course outline

Code: ICT221
Title: Software Development 1

Faculty of Arts, Business and Law
School of Business
Teaching Session: Semester 1
Year: 2017
Course Coordinator: Dr Mark Utting
Office: K2.02
Telephone: (07) 5459 4495
Email: utting@usc.edu.au
Consultation Times: As advised on Blackboard

1. What is this course about?

1.1 Course description
This course introduces you to object-oriented programming, which is the most widely-used modern programming paradigm. You will learn to design larger programs by structuring them into multiple classes, with a variety of relationships between those classes, such as association, composition, and inheritance. These techniques will be applied to the building of sophisticated graphical user interfaces (GUI).

1.2 Course content
Course overview
- Review of imperative programming
Inheritance & Polymorphism
- OO concepts to simplify program code
Exceptions & assertions
- Mechanisms for handling unexpected errors such as invalid user input
I/O handling for files
- Storing and retrieving data from files
GUI programming
- Creating software with Graphical User Interfaces (GUIs)
Abstract classes & interfaces
- Special generic classes to simplify program code
- Special structures to simplify program code
Event driven programming
- Mechanisms for undertaking tasks when events such as buttons being clicked occur
Creating user interfaces
- Detailed examination of user interface components such as buttons, lists, check boxes, menus, scroll bars and tabbed panes

2. Unit value
12 units
3. **How does this course contribute to my learning?**

<table>
<thead>
<tr>
<th>Specific Learning Outcomes</th>
<th>Assessment Tasks</th>
<th>Graduate Qualities</th>
</tr>
</thead>
<tbody>
<tr>
<td>On successful completion of this course you should be able to:</td>
<td>You will be assessed on the learning outcome in task/s:</td>
<td>Completing these tasks successfully will contribute to you becoming:</td>
</tr>
<tr>
<td>Creation of software.</td>
<td>1 and 2</td>
<td>Creative and critical thinkers.</td>
</tr>
<tr>
<td>Apply initiative to software development.</td>
<td>2</td>
<td>Empowered.</td>
</tr>
<tr>
<td>Apply communication skills to software development.</td>
<td>2</td>
<td>Engaged.</td>
</tr>
<tr>
<td>Apply software development knowledge and skills to problems.</td>
<td>1, 2 and 3</td>
<td>Knowledgeable.</td>
</tr>
</tbody>
</table>

4. **Am I eligible to enrol in this course?**

Refer to the *Coursework Programs and Awards - Academic Policy* for definitions of “pre-requisites, co-requisites and anti-requisites”

4.1 **Enrolment restrictions**
Nil

4.2 **Pre-requisites**
(ICT112 and ICT115) or SGD203

4.3 **Co-requisites**
Nil

4.4 **Anti-requisites**
Nil

4.5 **Specific assumed prior knowledge and skills**
Knowledge of simple imperative programming (loops, conditionals, functions, arrays) is assumed as a prerequisite of this course.

5. **How am I going to be assessed?**

5.1 **Grading scale**
Standard – High Distinction (HD), Distinction (DN), Credit (CR), Pass (PS), Fail (FL)
5.2 Assessment tasks

<table>
<thead>
<tr>
<th>Task No.</th>
<th>Assessment Tasks</th>
<th>Individual or Group</th>
<th>Weighting</th>
<th>What is the duration / length?</th>
<th>When should I submit?</th>
<th>Where should I submit it?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tutorial assessed exercises</td>
<td>Individual</td>
<td>10%</td>
<td>N/A</td>
<td>Weeks 1-10</td>
<td>In computer workshop</td>
</tr>
<tr>
<td>2</td>
<td>Programming exercises</td>
<td>Individual</td>
<td>40%</td>
<td>N/A</td>
<td>Weeks 9 and 13</td>
<td>Online via Blackboard</td>
</tr>
<tr>
<td>3</td>
<td>Final examination</td>
<td>Individual</td>
<td>50%</td>
<td>2 hours</td>
<td>Central examination period</td>
<td>In exam venue</td>
</tr>
</tbody>
</table>

Assessment Task 1: Tutorial assessed exercises

**Goal:** Design, document and create software.

**Product:** During tutorials 1 to 10 you will receive a small exercise to be completed during the tutorial. The tutorial exercises will be marked in the labs, and these will be used to determine the overall tutorial exercise grade. Each exercise must be completed and submitted prior to the specified deadline.

**Format:** Individual

**Criteria** Each tutorial exercise will be marked with a 0-5 grade corresponding to whether a reasonable attempt has been made by the specified deadline.

**Generic skill assessed**

<table>
<thead>
<tr>
<th>Problem solving</th>
<th>Skill assessment level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developing</td>
<td></td>
</tr>
</tbody>
</table>

**Generic skill assessed**

<table>
<thead>
<tr>
<th>Applying technologies</th>
<th>Skill assessment level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developing</td>
<td></td>
</tr>
</tbody>
</table>

Assessment Task 2: Programming exercises

**Goal:** Design, document and create software.

**Product:** Individual project incorporating design, documentation and programming.

**Format:** Individual

**Criteria** The assessment criteria will be supplied with the exercises.

**Generic skill assessed**

<table>
<thead>
<tr>
<th>Problem solving</th>
<th>Skill assessment level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developing</td>
<td></td>
</tr>
</tbody>
</table>

**Generic skill assessed**

<table>
<thead>
<tr>
<th>Applying technologies</th>
<th>Skill assessment level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developing</td>
<td></td>
</tr>
</tbody>
</table>

Assessment Task 3: Final examination

**Goal:** Demonstrate knowledge of course content.

**Product:** A two (2) hour closed book examination at the end of the semester comprising questions from the information obtained during the computer workshop activities, lecture material and additional readings specified during the lecture series.

**Format:** Individual

**Criteria** The examination assessment will be comprised of five short-answer questions, as well as multiple choice questions.

**Generic skill assessed**

<table>
<thead>
<tr>
<th>Problem solving</th>
<th>Skill assessment level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developing</td>
<td></td>
</tr>
</tbody>
</table>

**Generic skill assessed**

<table>
<thead>
<tr>
<th>Applying technologies</th>
<th>Skill assessment level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developing</td>
<td></td>
</tr>
</tbody>
</table>
5.3 Additional assessment requirements
Material essential to the course learning objectives will be covered, and each week will build upon topics from previous weeks. Moreover, as described previously, assessed tutorial exercise grades are an assessment item for this course.

Development Environment
Whilst students with their own computers can set-up the software development environment, no guarantee can be made that the environment will be the same as that set-up in the Computer Laboratories of USC. Students should not spend an inappropriate time setting up their own computers, as the USC computers are available to complete all assignments. Difficulties in using the environment on a personal computer will not constitute an excuse for late submission of any assessment items.

Plagiarism
In order to minimise incidents of plagiarism and collusion, this course may require that some of its assessment tasks, when submitted to Blackboard, are electronically checked through SafeAssign. This software allows for text comparisons to be made between your submitted assessment item and all other work that SafeAssign has access to.

Eligibility for Supplementary Assessment
Your eligibility for supplementary assessment in a course is dependent of the following conditions applying:
a) The final mark is in the percentage range 47% to 49.4%
b) The course is graded using the Standard Grading scale
c) You have not failed an assessment task in the course due to academic misconduct

5.4 Submission penalties
Late submission of assessment tasks will be penalised at the following maximum rate:
- 5% (of the assessment task’s identified value) per day for the first two days from the date identified as the due date for the assessment task.
- 10% (of the assessment task’s identified value) for the third day
- 20% (of the assessment task’s identified value) for the fourth day and subsequent days up to and including seven days from the date identified as the due date for the assessment task.
- A result of zero is awarded for an assessment task submitted after seven days from the date identified as the due date for the assessment task.

Weekdays and weekends are included in the calculation of days late. To request an extension you must contact your course coordinator to negotiate an outcome.

6. How is the course offered?

6.1 Directed study hours
On campus Lecture: 1 hour per week
On campus Computer Workshop: 2 hours per week

6.2 Teaching semester/session(s) offered
Semester 1
6.3 Course activities

<table>
<thead>
<tr>
<th>Teaching Week / Module</th>
<th>What key concepts/content will I learn?</th>
<th>What activities will I engage in to learn the concepts/content?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Directed Study Activities</td>
</tr>
</tbody>
</table>
| 1                      | Course overview  
Review of imperative programming.  
Introduction to Java                      | Creating programs         | As provided each week on Blackboard                           |
| 2                      | Introduction to Java  
Data, control structures, methods                                 | Program design exercises  | As provided each week on Blackboard                           |
| 3                      | Classes and Objects  
Data fields, methods, constructors, UML class diagrams                     | Program design exercises  | As provided each week on Blackboard                           |
| 4                      | Object-Oriented Thinking  
Relationships between classes, object-oriented design                                                            | Program design exercises  | As provided each week on Blackboard                           |
| 5                      | Inheritance & Polymorphism  
OO concepts to simplify program code                                                               | Program design exercises  | As provided each week on Blackboard                           |
| 6                      | Unit Testing  
JUnit and Test Driven Development (TDD)                                                      | Program design exercises  | As provided each week on Blackboard                           |

**Mid Semester Break**

| 7                      | File I/O and Exceptions  
Text files and streams in Java                                                                  | Program design exercises  | As provided each week on Blackboard                           |
| 8                      | Abstract Class and Interfaces  
Special generic classes to simplify program code                                                   | Program design exercises  | As provided each week on Blackboard                           |
| 9                      | Graphical User Interfaces (GUI)  
GUI Programming  
JavaFX basics                                                                                            | Program design exercises  | As provided each week on Blackboard                           |
| 10                     | Event-Driven Programming  
Call-back methods                                                                                   | Program design exercises  | As provided each week on Blackboard                           |
| 11                     | Advanced User Interfaces  
JavaFX controls and layouts                                                                         | Program design exercises  | As provided each week on Blackboard                           |
| 12                     | User Interface Design Patterns  
Model-View-Controller (MVC)                                                                          | Program design exercises  | As provided each week on Blackboard                           |
| 13                     | Course Summary  
Activities from all above weeks                                                                  | Program design exercises  | As provided each week on Blackboard                           |

**Study Period**

**Central Examination Period**

**End of Semester Break**

Please note that the course activities may be subject to variation.
7. What resources do I need to undertake this course?

7.1 Prescribed text(s)

<table>
<thead>
<tr>
<th>Author</th>
<th>Year</th>
<th>Title</th>
<th>Publisher</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y. Daniel Liang</td>
<td>2015, 10th ed</td>
<td>Introduction to Java Programming, Brief Version</td>
<td>Pearson</td>
</tr>
</tbody>
</table>

7.2 Required and recommended readings
Lists of required and recommended readings may be found for this course on its Blackboard site. These materials/readings will assist you in preparing for tutorials and assignments, and will provide further information regarding particular aspects of your course.

7.3 Specific requirements
Nil

7.4 Risk management
Health and safety risks have been assessed as low. It is your responsibility to research and understand risks of specific courses and to review the USC’s health and safety principles by viewing the online induction training for students.

8. How can I obtain help with my studies?
In the first instance you should contact your tutor, then the Course Coordinator. Additional assistance is provided to all students through Peer Advisors and Academic Skills Advisors. You can drop in or book an appointment. To book: Tel: +61 7 5430 2890 or Email: studentcentral@usc.edu.au

9. Links to relevant University policies and procedures
For more information on Academic Learning & Teaching categories including:
- Assessment: Courses and Coursework Programs
- Review of Assessment and Final Grades
- Supplementary Assessment
- Administration of Central Examinations
- Deferred Examinations
- Student Academic Misconduct
- Students with a Disability


10. General enquiries
In person:
- **Sippy Downs** - Student Central, Ground Floor, Building C
- **USC SouthBank** - Student Central, Building B, Ground floor (level 1)
- **USC Gympie** - Student Central, 71 Cartwright Road, Gympie
- **USC Fraser Coast** - Student Central, Building A
Tel: +61 7 5430 2890
Email: studentcentral@usc.edu.au