Unit Outline: KXC151 Programming and Problem Solving

September 2006 - January 2007
Hangzhou, China

Prerequisites
None
Corequisites
None
Unit Weight
12.5% of one academic year
Unit Coordinator
Philip Uren
Lecturing Staff
Dr Ding Weilong
Email: wlding@ustc.edu
Scheduled Teaching Sessions
Lectures: 3hr/week (equivalent)
Tutorials: 1hr/week (equivalent)
Unit Website
The unit website is accessed from http://www.utas.edu.au/coursesonline/. You will need to use your University of Tasmania email pop account username and password to log on to the WebCT system. Once authenticated by the system your personalised MyWebCT area will be displayed. It contains links to the websites that you have permission to access - including the website for this unit.

This unit is Web Dependent: content. This means that you will need to use the Web for this unit. The unit website contains unit information and resources.
If you are not able to access the unit website, please contact the technical staff at ZUT.

University Website
Information and Resources for 'Current Students' are available on the university website at: http://www.utas.edu.au/students/

Provider
School of Computing - Faculty of Science, Engineering, and Technology.

OVERVIEW

Introduction
Students learn to use a high level language such as Java to write programs which solve problems defined by a program specification. They master fundamental concepts relating to imperative, object-based programming and are introduced to concepts relating to graphical user interfaces and event driven programs. Students are required to demonstrate syntactic, logical and strategic knowledge of the programming constructs introduced in the unit. They are expected to use systematic processes to plan, document, debug and test their programs. Programming exercises are introduced in the context of small problems.

Warning on Over-confidence: Some students who have done a considerable amount of home or school computing may think that they are already expert computer programmers. This is extremely unlikely, as most self-taught or uncorrected programmers have picked up bad habits which are inappropriate in professional programming, and may have major gaps in their understanding of concepts. Please bear in mind that practising computing at a professional level is very different from practising it as a hobby. Experience has shown that very few students who have studied computing at school are so good that they can treat programming units lightly.

Prior Learning
This unit does not have any formal prerequisites. However, students need to navigate the web site for the unit and use a keyboard to write their programs.

Students who do not have the basic skills required to "surf" the web or to use a computer keyboard (at about the level required to use a simple word processor) will need to spend extra time early in the semester learning these (simple) skills.

Learning Outcomes
On successful completion of this unit, you will be able to:

1. Write, compile, and run Java programs that contain statements of the types introduced in the unit (see unit content).
   This demonstrates syntactic knowledge of programming constructs.
2. Understand the effect of Java statements of the types introduced in the unit (see unit content).
This demonstrates conceptual knowledge of programming constructs.

3. Analyse a problem specification and plan and produce a program which is a solution to the problem and uses Java statements of the types introduced in the unit (see unit content). This demonstrates strategic knowledge of programming constructs.

4. Use standard techniques to document work. This will include:
   - Appropriate documentation of the programs written during the semester.
   - Formal recording of aspects of activities throughout the process of software development.
   - Production of a record book which documents the activities undertaken in this unit during the semester and is a suitable aide memoire for use in the formal examination.

---

**Unit Content**

**Introduction:**
- Unit introduction
- Programming terms & tools
- Computing terms & tools
- Solving problems with computers

**Data Storage:**
- Primitive types
- Objects

**Objects of prewritten classes:**
- Introduction to objects
- Object methods

**Flow of control:**
- Branches
- Multi-way branching
- Loops
- Nesting flow of control

**Extending existing classes:**
- Writing methods
- Testing methods
- Method parameters and return values

**Creating new classes:**
- Planning
- Implementation

**Documenting programs:**
- Purpose of documentation
- Internal and external documentation

**Structured data - arrays:**
- Declaring & filling arrays
- Using arrays
- Arrays - sorting algorithms
- Arrays - searching algorithms

**Graphical User Interfaces (GUI):**
- Drawing a GUI
- Adding components to a GUI
- Making a GUI respond to events

**Types of errors in programs:**
- Run time errors - exceptions
- Handling exceptions

**Recursion:**
- Concepts
- Implementation

**Revision:**
- Concepts summarised
Exam techniques

For more information see the section titled 'Content' on the unit website.

Generic Skills

The university has defined a set of generic graduate attributes expected in its graduates. [http://www.utas.edu.au/policy/subject.html#graduates](http://www.utas.edu.au/policy/subject.html#graduates) Your course is designed to enable you to develop generic skills that are valued in, and expected of, graduates. These are skills that you will need to develop over time. Hence you are encouraged to look for opportunities, as you study each unit, to reflect on and improve these skills.

In this unit these skills are specifically targeted:
- **Knowledge**: Students will have the opportunity to begin the acquisition of the knowledge and understanding of computer programming which is a fundamental requirement for all professionals in information technology.
- **Problem-solving skills**: Students learn and practise the fundamental skills needed when attempting to write a computer program that correctly solves a problem that has been set.

LEARNING AND TEACHING

Approach to Learning

You are expected to spend about 130 hrs studying in this unit - this includes attendance at scheduled teaching sessions. (For a 13 week semester this is, on average, 10 hr/wk.) This is the amount of study time that the ‘typical’ student will need to reach the level of competence and understanding required to fulfil the unit objectives.

You are expected to:

- attend all scheduled lectures and tutorials, unless otherwise notified by the unit coordinator
- prepare for, and actively participate in lectures and tutorials
- complete the assigned learning tasks
- review what has been learnt
- complete assessment items and submit them on time
- access and be familiar with the information and resources available on the unit website
- seek help from teaching staff if you have any questions or difficulties in studying this unit

You are encouraged to read the university’s [Code of Conduct for Teaching and Learning](http://www.utas.edu.au/tl/policies/codes.html). Part A describes the ‘Responsibility of the University to Students’ and part B describes the ‘Responsibilities of Students to the University’.

Schedule

See the 'Schedule' section on the unit website.

Teaching and Support Staff

**Teaching Staff**

**Unit Coordinator:**

Philip Uren  
E-Mail: puren@utas.edu.au

**Lecturing Staff**

Dr Ding Weilong  
Email: wlding@ustc.edu

**School Help Desk**

Contact technical staff at ZUT for information about accessing and using the Computer labs.

**University Services and Support**

The University has staff available to assist you, such as the:

- Learning Development Advisor  
- Student Counselor  
- Careers Advisor  
- Disability Officer

For more information and contact details see the Services and Support section on the University ‘Current Students’ web page. [http://www.utas.edu.au/students/](http://www.utas.edu.au/students/)

Resources

**Unit Website**

The unit website contains unit information and resources. This includes administrative information, teaching materials, lecture overheads, tutorial exercises, and other resources. Students will have access to a copy of the unit website that can be installed on their own
machines and used without logging on to the internet.

**Prescribed Text**


**Readings**

Students are not required (and are unlikely to need) to use any resources other than those provided in the text book and the unit materials. If students wish to read more about program development using the Java programming language, there are many standard texts and freely available web sites with relevant information. Students using such resources should be aware that there are many approaches to introducing learners to programming in Java, it is possible that reading a book or web site that takes a different approach from the one used in this unit may increase rather than decrease confusion.

**Software**

The software that you will need to access the unit website and to study this unit, including general purpose software such as word processors, is provided on the computers in the computing labs. If you intend to use software on other computers please check that the versions are compatible.

Students will be provided with the necessary files and documentation to install the correct version of Java (and associated resources) on their own machines.

---

### ASSESSMENT

| Assessment Items | Item 1 | Title: Assignment 1  
|                  | Type: In-Semester - individual assignment  
|                  | Weighting: 12%  
|                  | Due: 5pm Friday 3rd November  
|                  | This will require students to:  
|                  | - write a small Java program with a single class  
|                  | - provide some specified items of documentation for the program and the process they followed in producing the program.  

| Item 2 | Title: Assignment 2  
| Type: In-Semester - individual assignment  
| Weighting: 18%  
| Due: 5pm Friday 29th December  
| This will require students to write code that will form part of a Java program which consists of several interacting classes. The code that students write will be expected to  
| - perform correctly when integrated with prewritten code provided as part of the program specification.  
| - consist of several methods  
| - show good programming style  
| - conform with the programming standards and naming conventions expected in this unit  
| - Provide some specified items of documentation for the program and the process they followed in producing the program  

| Item 3 | Title: Final Examination  
| Type: Formal Examination  
| Weighting: 70%  
| Due: University Examination Period  

This will consist of 2 sections.
- Section A - Carries 1/6 of the marks and consists of multiple choice questions.
- Section B - Carries 5/6 of the marks. Students will be required to answer 5 "long" questions. Each question will require the student to demonstrate their ability to complete some part of a programming and or problem solving exercise.

**NOTE:** The only materials that students will be permitted to take into the formal examination will be the Record book which they have produced during the course of the semester.

See the 'Assessment' section in unit website for more detailed information about assessment items.

**In-Semester Assessment**

Unless specifically stated in the specification of the assessment item provided on the unit website, it is required that:
- work submitted by a student is the work of that student alone OR
- where the assessment item is to be completed by a group of students, the work submitted by the group of students is the work of that group of students alone.

**Plagiarism**

Plagiarism is a form of cheating. It is taking and using someone else's thoughts, writings or inventions and representing them as your own, for example:
- using an author's words without putting them in quotation marks and citing the source;
- using an author's ideas without proper acknowledgment and citation; or
- copying another student's work.

**If you have any doubts about how to refer to the work of others in your assignments, please consult your lecturer or tutor** for relevant referencing guidelines, and the academic integrity resources on the web at http://www.utas.edu.au/tl/supporting/academicintegrity/index.html.

The intentional copying of someone else's work as one's own is a serious offence punishable by penalties that may range from a fine or deduction/cancellation of marks and, in the most serious of cases, to exclusion from a unit, a course or the University. Details of penalties that can be imposed are available in the Ordinance of Student Discipline – Part 3 Academic Misconduct, see http://www.utas.edu.au/policy/subject.html#students.

The University reserves the right to submit assignments to plagiarism detection software, and might then retain a copy of the assignment on its database for the purpose of future plagiarism checking.

**Referencing**

The university document on plagiarism contains information about referencing the work or ideas of others. (See http://www.utas.edu.au/plagiarism/.) The preferred text referencing systems for the School is the Harvard system (also referred to as the author-date system).

**Submissions**

The details of the submission method (paper, electronic or other) for each assignment will be supplied in a separate assignment specification sheet. All in-semester assignment submissions (including electronic submissions) are to include an Assignment Cover Sheet which includes a statement confirming that the submission is your own work. If this undertaking is not signed, the assignment will not be marked. The Assignment Cover Sheet is available on the School’s web site http://www.comp.utas.edu.au/app/studyresources.jsp.

**Extensions**

Assessment items will not be accepted after the due date except under the conditions stated in the school policy on late assessment. http://www.comp.utas.edu.au/app/late_assess.jsp

**Formal Examination**

The formal examination will be held at ZUT, Hangzhou, and is conducted by the University Registrar.

**Final Grade**

Overall assessment will be based on the student's performance throughout the semester as well as in a formal examination. In order to achieve a pass (or better) result, a student must obtain:

1. at least 45% of the total mark for in-semester assessment items
2. at least 45% of the mark for the formal examination
3. at least 50% of the overall mark

In order to comply with the benchmarks set by the Faculty of Science, Engineering & Technology for distribution of grades in units, both the in-semester and examination marks that students obtain may be adjusted either upwards or downwards. See http://fcms.its.utas.edu.au/scieng/scieng/policies.asp for details of the Faculty Assessment Guidelines.

In order to comply with the benchmarks set by the Faculty of Science, Engineering & Technology for distribution of grades in units, the results that students obtain may be scaled (either upwards or downwards). See http://fcms.its.utas.edu.au/scieng/scieng/policies.asp for details of the Faculty Assessment Guidelines.

Passing grades will be awarded based on the AVCC guidelines:

- PP at least 50% of the overall mark but less than 60%
- CR at least 60% of the overall mark but less than 70%
- DN at least 70% of the overall mark but less than 80%
- HD at least 80% of the overall mark

The maximum mark awarded to a student who fails the unit will be 44.

For more information, including other grades such as Supplementary and Terminating grades, see the School of Computing's guidelines for assessment - available at: http://www.comp.utas.edu.au/app/assess.jsp