

# **COMP1500/COMP7901**

## **Introduction to Programming/ Software Engineering**

**Course Profile**  
**Semester 2**  
**2004**

**St.Lucia Campus**

School of Information Technology and Electrical Engineering  
Faculty of Engineering, Physical Sciences and Architecture



**THE UNIVERSITY  
OF QUEENSLAND**

Template design by Learning Resources Development Unit, Teaching and Educational Development Institute, The University of Queensland:

Copyrighted materials reproduced herein on behalf of The University of Queensland are used either under the provisions of the *Copyright Act 1968* as amended, or as a result of application to the copyright owner.

This material may not be reproduced in any manner except for the purposes of individual study.

© The University of Queensland 2004

# Course profile

## Course details

### Aim

### Assumed background

### Learning outcomes

### Graduate attributes

### Resources

- Study guide
- Required textbook
- Assessment and assignment booklets
- Website
- Newsgroup
- Computer facilities
- Notices
- Consultation

### About the Course

- Lecture times each week

### Study chart

### Assessment

- Lab assessment
- Assignments
- Practical exam
- Multiple-choice exam
- How a grade is awarded
- Checking your marks
- Academic merit, plagiarism, collusion and other misconduct

### How to successfully study this course

### Getting help

- Whom to see about problems
- Disability support

## Course details

undergrad code: <b>COMP1500</b>	<b>Introduction to programming</b>
postgrad code: <b>COMP7901</b>	<b>Software engineering</b>
Units: 2	Study hours/week: 12
Year: 2004	Semester 2
<b>Lecturing staff:</b>  Roger Duke (course coordinator) School of Information Technology and Electrical Engineering, Room 78-324 Tel: 3365 3335, e-mail: rduke@itee.uq.edu.au	
<b>Course e-mail:</b>  <b>comp1500@itee.uq.edu.au</b> or <b>comp7901@itee.uq.edu.au</b>	
Please send all course-related email to the appropriate one of these two addresses.	

## Aim

The main aim of this course is to explore the art and science of software engineering by taking an experiential approach to learning and using the Java programming language. Independent and self-paced learning is combined with traditional-style lectures and supervised lab sessions. Software engineering can be a tremendously challenging and exciting experience, and programming in Java in particular can be a lot of fun. This course has been designed to be fun and exciting while providing the challenge of learning the fundamentals of programming.

## Assumed background

Senior mathematics B or equivalent experience is recommended. While mathematical sophistication is not required, basic mathematical skills will assist when capturing logical relationships or constructing algorithms.

## Learning outcomes

On successful completion of this course, you will be able to:

- demonstrate a thorough working knowledge of the Java programming language, and be able to apply these concepts to write Java programs
- develop a good working knowledge of a Java programming environment (e.g. the Kawa programming environment)
- recognise some of the software algorithms traditionally used to implement basic procedures, and apply these algorithms within your own code when required
- understand and apply a broad range of object-oriented concepts when programming in Java
- recognise the crucial role of architectural structures within software systems, and apply common examples from existing systems when constructing your own applications and applets
- identify some basic patterns that frequently occur in software code and design and apply these, at an introductory level, in your own applications and applets
- understand how to program graphical applications and applets and apply the basic techniques for creating animations using Java.

Each chapter of the textbook *Java Genesis* also details its own expected outcomes in terms of knowledge and skills that should result from a detailed study of that chapter. At the end of each chapter there is a self check which can be used to measure personal achievement.

## Graduate attributes

A University graduate is expected to have five attributes: knowledge, communication skill, creativity, judgment and civic understanding. Each course must show how it contributes to these ends. On successful completion of this course you should have:

- *Knowledge*: a thorough knowledge of the Java programming language; knowledge of the role of software patterns and complex data structures in building software; and knowledge of the fundamental programming concepts that underlie the Java language.
- *Judgment*: the ability to analyse problems from an algorithmic point of view; the rudimentary ability to evaluate programs as superior or inferior (simple/complicated; efficient/inefficient).
- *Creativity*: the ability to solve programming problems; to adapt similar problems and solutions to new requirements; to develop alternative solutions to those given; to explore a wide range of programming problems and solutions.
- *Communication skill*: to communicate effectively with a machine in a very rigid language.
- *Civic understanding*: the School demands good conduct in computing laboratories and honest conduct in submitting work.

# Resources

## Study guide

This course involves studying the text book *Java Genesis*, attending lectures, attending lab sessions, completing and submitting assessment and assignment problems, attending a practical exam and finally, attending a multiple-choice exam. The study chart given later in this document gives an indication of the progress you should try to maintain.

## Required textbook

The set text for this course is:

Duke, R. and Salzman, E., *Java Genesis*, 2<sup>nd</sup> Edition, Pearson Education Australia, 2004.

You can purchase this textbook from the UQ bookshop.

A CD-ROM containing the latest Java and Kawa installation code can be borrowed from the School office (level 2, building 78).

## Assessment and assignment booklets

These can be found on the course website.

## Website

The course website can be found at:

<http://www.itee.uq.edu.au/~comp1500/>

or

<http://www.itee.uq.edu.au/~comp7901/>

It contains the course profile, assignments and lab assessments. As the semester progresses lecture slides, specimen exams, quick quizzes, challenge problems and student marks, as well as up to the date course notices, will be added to the website. **You should check this site regularly.**

## **Newsgroup**

You can use the newsgroups `uq.itee.comp1500` or `uq.itee.comp7901` to communicate with other students about issues related to this course.

## **Computer facilities**

The course requires the use of computers that support Java. These computers are available in the School labs to all enrolled students. Should you wish to do so, you may use your own computer resources for the course. However, you must understand that you do so at your own risk. The University is not responsible for any hardware or software problems that arise as a result. School staff can not provide help or advice regarding your personal hardware/software configuration. Furthermore, you must verify that all computer-based assignments that you plan to submit will execute correctly in the School laboratories. Computer-based assignments that do not execute correctly in the School labs will not be accepted.

## **Notices**

Information of interest to students taking this course will be posted on the notice boards outside the computing labs (level 1, building 78). Notices will also be added to the course website.

## **Consultation**

The lecturer will be available at designated times during the week to answer on a one-on-one basis any questions regarding course material. The times and locations will be announced.

To get assistance with any programming task, students should consult with a lab tutor during *any* of the practical sessions.

## About the course

This course consists of the following 11 topics taken from the *Java Genesis* textbook.

- Exploring Java
- Basic programming constructs
- Control constructs
- Arrays
- Methods
- Objects and classes
- Inheritance
- Graphics and event handling
- Graphical components
- Exceptions and files
- Applets

The course consists of

- 24 lectures (2 each week for the first 12 teaching weeks). Lectures focus on demonstrating concepts and techniques needed to tackle the problems in *Java Genesis*, as well as assessment and assignment tasks
- 36 hours of supervised lab work. Students are allocated 3 hours each week of supervised practical work in the computing laboratory where immediate advice and help with problems can be obtained.

The course is designed for independent learning, so you may work on the problems elsewhere (e.g. at home) if you prefer. You are also encouraged to spend 5–6 additional hours each week outside of the allocated lab time working on the problems.

### Lecture times each week

**Lecture 1: 12 noon to 12:50 pm Tuesday**  
**in room 23-1 (Abel Smith Lecture Theatre)**

**Lecture 2: 12 noon to 12:50 pm Friday**  
**in room 50-1 (Hawken Engineering Building)**

## Study chart: Semester 2 2004

This timetable gives you an indication of the progress you should try to maintain.

Teaching Week	Date (Mon)	<i>Java Genesis</i> Study task	Assessment task (deadline)	Lectures
1	July 26	Ch. 1: Getting started Ch. 2: Exploring Java	Lab Assessment 1 (week 5)	1: Introduction 2: Objects and messages
2	Aug 2	Ch. 3: Basic programming constructs	Lab Assessment 2 (week 5)	1: Arithmetic operations 2: Iteration, for-loops
3	Aug 9	Ch. 4: Control constructs	Lab Assessment 3 (week 5)	1: Coding algorithms 2: Control constructs
4	Aug 16	Ch. 5: Arrays (Section 5.4 is optional)	<b>Assignment 1 (August 20, week 4)</b>	1: Introducing arrays 2: Working with arrays
5	Aug 23	Ch. 6: Methods	Lab Assessment 4 (week 8)	1: Coding methods 2: Exploring methods
6	Aug 30	Ch. 7: Objects and classes	Lab Assessment 5 (week 8)	1: Creating objects 2: Case study
7	Sept 6		<b>Assignment 2 (September 10, week 7)</b>	1: Objects and methods 2: Case study
8	Sept 13	Ch. 8: Inheritance	Lab Assessment 6 (week 10)	1: Intro to inheritance 2: A day at the zoo
9	Sept 20	Ch. 9 Graphics and event handling	Lab Assessment 7 (week 10)	1: Displaying graphics 2: Handling events
	Sept 27	(mid-semester break)		
10	Oct 4	Ch. 10 Graphical components	Lab Assessment 8 (week 12)	1: Hunting the treasure 2: Building a calculator
11	Oct 11	Ch. 11: Exceptions and files	<b>Assignment 3 (October 15, week 11)</b>	1: Exceptions 2: File I/O
12	Oct 18	Ch. 12: Applets	Lab Assessments 9 and 10 (week 12)	1: Applets 2: The sample exam
13	Oct 25		<b>Practical exam in labs</b>	<b>no lectures</b>

# Assessment

(in the recommended order of study)

Assessment	Marks	Deadline
Lab Assessment 1	2	week 5
Lab Assessment 2	2	week 5
Lab Assessment 3	2	week 5
<b>Assignment1</b>	<b>10</b>	<b>August 20 (week 4)</b>
Lab Assessment 4	2	week 8
Lab Assessment 5	2	week 8
<b>Assignment 2</b>	<b>10</b>	<b>Sept 10 (week 7)</b>
Lab Assessment 6	2	week 10
Lab Assessment 7	2	week 10
Lab Assessment 8	2	week 12
<b>Assignment 3</b>	<b>10</b>	<b>October 15 (week 11)</b>
Lab Assessment 9	2	week 12
Lab Assessment 10	2	week 12
<b>Practical exam</b>	<b>30</b>	<b>week 13</b>
<b>Multiple-choice exam</b>	<b>20</b>	<b>exam period</b>

The 10 lab assessments are worth 20 marks in total (2 marks each).

The 3 assignments are worth 30 marks in total (10 marks each).

The practical exam is worth 30 marks.

The multiple-choice exam is worth 20 marks.

(This gives 100 marks in total for the assessment.)

## Lab assessment

The Lab Assessment booklet contains 10 programming problems each worth 2 marks. To get full marks you will need to demonstrate and explain your solution to the problem to a tutor in the computing lab during an allocated lab session before the end of the teaching week deadline listed in the table above.

**Lab assessments will not be marked after the deadline week.**

The assessment criteria for each Lab problem are based on the learning outcomes of the relevant chapter(s) of Java Genesis and will be assessed on that basis. Marks for a lab assessment will generally be awarded on the basis of 2 or zero. You will be able to get tutor assistance if you are having difficulty completing a lab assessment problem.

## Assignments

The Assignments booklet contains three assignment problems each worth 10 marks. Each assignment involves preparing and submitting code. These assignments fall at strategic points in the study of *Java Genesis* and can be seen as a natural extension of the chapters. Each assignment problem is designed to test the overall learning outcomes for a set of chapters of *Java Genesis*. They also broaden the context and incorporate wider issues.

Consult the Assignments booklet for details of the assessment criteria for each assignment problem. In general terms, the mark awarded will depend on your understanding of the concepts covered and your ability to apply them in a new context.

Each assignments must be submitted before the deadline. Details of submission are given in the assignments booklet.

**Late assignments will not be accepted.**

## Practical exam

An open-book practical exam worth 30 marks will be held in the labs during week 13, the final teaching week. This exam is designed to test practical knowledge and skills learned throughout the course.

You will be permitted to take the text book *Java Genesis* as well as any written notes into the practical exam. However, you will not be permitted to take floppy disks into the exam.

## Multiple-choice exam

A multiple-choice exam worth 20 marks will be held during the exam period (June 17 to June 29). This will be a closed-book exam, i.e. you will not be permitted to take notes or other course material into the exam. This exam is designed to test your understanding of the semantics (i.e. the meaning) of Java code.

## How a grade is awarded

You will pass this course (i.e. get a grade of 4 or better) if

- you get 50 marks or more in total for the assessment (i.e. if the total of your marks for the lab assessment, the assignments, the practical exam and the multiple-choice exam is at least 50)
- **and** you get
  - **either** 15 marks or more for the practical exam
  - **or** 25 marks or more for the total of the practical and multiple-choice exam.

The implication of this is

- you will not pass the course if you do not pass the exams, regardless of how well you do in the continuous assessment (i.e. the lab assessment and assignments);
- you will pass the exams if you get 15 or more marks for the practical exam or get 25 or more marks for the total of the practical and multiple-choice exams.

Here are the details of how the passing grades are determined.

Let  $E$  be the mark out of 50 for the exams, i.e. the sum of the practical and multiple-choice exams, and let  $T$  be the total mark out of 100 for all assessment, i.e. the sum of  $E$ , the lab assessments and the assignments.

- A grade of 7 will be awarded if  $E \geq 45$  and  $T \geq 90$ .
- A grade of 6 will be awarded if  $E \geq 40$  and  $T \geq 80$  and a grade of 7 is not awarded.
- A grade of 5 will be awarded if the exams are passed and  $T \geq 70$  and a grade of 6 or better is not awarded.
- A grade of 4 will be awarded if the exams are passed and  $T \geq 50$  and a grade of 5 or better is not awarded.

## Checking your marks

Your marks recorded for the various components of the assessment will be posted on the website. The marks on the website will be updated once a week. You should check the website regularly to ensure your marks have been recorded accurately. In particular:

- you must report any errors with any lab assessment mark within 10 days of that assessment being marked by a tutor;
- if you wish to dispute your mark for any assignment this must be done within 7 days of the mark being posted on the website.

## Academic merit, plagiarism, collusion and other misconduct

You are encouraged to discuss your assessment and assignment problems with other students in the class. However, care must be taken not to exceed 'reasonable' boundaries when collaborating with others on such material.

The School and the wider academic community in general takes academic integrity and respect for other persons and property very seriously.

In particular, the following behaviour is unacceptable:

- Submission of plagiarised work, i.e. work that contains content taken from an unacknowledged source.
- Engaging in collusive behaviour, i.e. inappropriate working together with other students where individual work is required, or working with people outside your team where team work is required.
- Copying work done by other students.
- Failing to adhere to the School's regulations concerning behaviour in laboratories, in particular occupational health and safety regulations.

Penalties for engaging in unacceptable behaviour can range from cash fines or loss of grades in a course, through to expulsion from the University.

You are also reminded that submission of work without academic merit, i.e. work that adds little or nothing to material available from reference sources such as textbooks, websites, etc., *even where this is appropriately acknowledged*, will be marked accordingly and may result in a failing grade.

You are required to read and understand the School Statement on Misconduct, available on our website at:

**<http://www.itee.uq.edu.au/about/student-misconduct.jsp>**

If you have any questions concerning this statement, please contact your lecturer in the first instance.

## How to successfully study this course

In this course we use a problem-based learning approach that encourages self-directed exploration. In fact, *exploration* is one of the key themes. We encourage you to:

- *Explore*: be curious and look for new things to learn. If you find something interesting, investigate it and try to understand it and learn about it.
- *Experiment*: try new things and see what happens. Form hypotheses about why code works; make changes to code and observe what happens; based on your observation, either confirm or change your hypotheses and continue to experiment.
- *Experience*: sometimes the best way to understand code is to run it, i.e. to experience it.
- *Extend*: when you have learned something, try to extend your knowledge by linking it with other things you have learned.

This is a problem-based course — the problems are more important than the textual material. Naturally, the textual material needs to be read, but only by doing the problems will the art of Java programming be mastered.

The textbook *Java Genesis* is your detailed learning guide for this course. You should keep up with the problem exercises in the text and complete the lab assessments, getting assistance from tutors in the lab when you need it. In lectures you will see demonstrations of related code and design features. Using the newsgroup available through the website you can discuss problems and issues with other students.

We wish you success. Feedback is most welcome!

## Getting help

### Whom to see about problems

Tutors are allocated to the scheduled laboratory sessions. They are an invaluable resource in times of difficulty. Even if you aren't allotted to a particular laboratory session, the tutors are available for help. If difficulties persist, you should see a lecturer during consultation time (see lecturer contact details at the beginning of this profile).

If you send an email (`comp1500@itee.uq.edu.au` or `comp7901@itee.uq.edu.au` are the course addresses) be sure to include your name and student number. Including this information is important because of the large number of students taking this course. We regret that emails not containing this information will not be answered.

Every endeavour will be made to answer emails but no guarantee can be given.

### Disability support

Any student with a disability who may require alternative academic arrangements in this course is encouraged to seek advice at the commencement of the semester from a Disability Adviser at Student Support Services.