



MONASH University

**CPE4001**  
**Distributed programming in Java**

**Unit guide**

**Semester 2, 2008**

*Last updated : 02 Jul 2008*

# Table of Contents

<u>CPE4001 Distributed programming in Java - Semester 2 , 2008</u> .....	1
<u>Unit leader</u> :.....	1
<u>Lecturer(s)</u> :.....	1
<u>Caulfield</u> .....	1
<u>Introduction</u> .....	2
<u>Unit synopsis</u> .....	2
<u>Learning outcomes</u> .....	2
<u>Workload</u> .....	2
<u>Unit relationships</u> .....	2
<u>Prerequisites</u> .....	2
<u>Relationships</u> .....	3
<u>Continuous improvement</u> .....	4
<u>Student Evaluations</u> .....	4
<u>Unit staff - contact details</u> .....	5
<u>Unit leader</u> .....	5
<u>Lecturer(s)</u> :.....	5
<u>Teaching and learning method</u> .....	6
<u>Tutorial allocation</u> .....	6
<u>Communication, participation and feedback</u> .....	6
<u>Unit Schedule</u> .....	6
<u>Unit Resources</u> .....	8
<u>Prescribed text(s) and readings</u> .....	8
<u>Recommended text(s) and readings</u> .....	8
<u>Required software and/or hardware</u> .....	8
<u>Equipment and consumables required or provided</u> .....	8
<u>Study resources</u> .....	8
<u>Library access</u> .....	9
<u>Monash University Studies Online (MUSO)</u> .....	9
<u>Assessment</u> .....	10
<u>Unit assessment policy</u> .....	10
<u>Assignment tasks</u> .....	10
<u>Examinations</u> .....	11
<u>Assignment submission</u> .....	12
<u>University and Faculty policy on assessment</u> .....	13
<u>Due dates and extensions</u> .....	13
<u>Late assignment</u> .....	13
<u>Return dates</u> .....	13
<u>Plagiarism, cheating and collusion</u> .....	13
<u>Register of counselling about plagiarism</u> .....	14
<u>Non-discriminatory language</u> .....	14
<u>Students with disabilities</u> .....	14
<u>Deferred assessment and special consideration</u> .....	14

# **CPE4001 Distributed programming in Java - Semester 2 , 2008**

## **Unit leader :**

Michael Smith

## **Lecturer(s) :**

### **Caulfield**

- Michael Smith

## Introduction

### Unit synopsis

This subject focuses on the development of distributed and concurrent software applications, with Java as the teaching language.

Concurrent programming primitives and concepts: threads, monitors, safety and liveness, control policies, resource sharing. Application of concurrent techniques in distributed application designs: communications, coordination, reuse and fault tolerance. Design and implementation issues and techniques of distributed applications: synchronous and asynchronous communications, message passing, naming and trading of components, atomic and composite objects, object migration and replications. Enabling techniques: platform and language heterogeneity, infrastructure models for interoperability (such as Java/RMI), object interface definition, remote operation invocation. Distributed patterns. Service discovery and lookup, leases and transactions of resources. Distributed events. Case studies of distributed programming paradigms and their applications (e.g. JINI, JavaSpaces).

### Learning outcomes

At the completion of this subject, students will be able to:

- write distributed and concurrent software in Java;
- understand the concepts and characteristics of distributed and concurrent software;
- design distributed software applications using typical distributed software architectures;
- identify and evaluate common distributed and concurrent software designs.

### Workload

Student workload commitments are:

- two-hour lecture and
- two-hour tutorial (or laboratory) (requiring advance preparation)
- a minimum of 2-3 hours of personal study per one hour of contact time in order to satisfy the reading and assignment expectations.
- You will need to allocate up to 5 hours per week in some weeks.

### Unit relationships

#### Prerequisites

Before attempting this unit you must have satisfactorily completed an undergraduate qualification in computing or closely related discipline is required.

This is an advanced Java programming unit. Previous knowledge of Java programming is desired. The following is assumed knowledge and will not be taught in the unit: An understanding of object-oriented concepts, terminology, analysis, design and testing. Use of object-oriented models such as UML diagrams. Knowledge of core Java packages. Some exposure to multithreading. Knowledge of all Java language constructs such as loops, conditionals, methods, classes, inheritance, interfaces, arrays and the Java Collections Framework, exception handling, file input and output.

A rudimentary knowledge of JDBC and SQL is also desirable.

CPE4001 Distributed programming in Java - Semester 2 , 2008

Students without appropriate programming skills and background knowledge are expected to engage in extra private studies.

## **Relationships**

CPE4001 is an elective unit in the Masters coursework degrees

## Continuous improvement

Monash is committed to 'Excellence in education' and strives for the highest possible quality in teaching and learning. To monitor how successful we are in providing quality teaching and learning Monash regularly seeks feedback from students, employers and staff. Two of the formal ways that you are invited to provide feedback are through Unit Evaluations and through Monquest Teaching Evaluations.

One of the key formal ways students have to provide feedback is through Unit Evaluation Surveys. It is Monash policy for every unit offered to be evaluated each year. Students are strongly encouraged to complete the surveys as they are an important avenue for students to "have their say". The feedback is anonymous and provides the Faculty with evidence of aspects that students are satisfied and areas for improvement.

## Student Evaluations

The Faculty of IT administers the Unit Evaluation surveys online through the my.monash portal, although for some smaller classes there may be alternative evaluations conducted in class.

If you wish to view how previous students rated this unit, please go to <http://www.monash.edu.au/unit-evaluation-reports/>

Over the past few years the Faculty of Information Technology has made a number of improvements to its courses as a result of unit evaluation feedback. Some of these include systematic analysis and planning of unit improvements, and consistent assignment return guidelines.

Monquest Teaching Evaluation surveys may be used by some of your academic staff this semester. They are administered by the Centre for Higher Education Quality (CHEQ) and may be completed in class with a facilitator or on-line through the my.monash portal. The data provided to lecturers is completely anonymous. Monquest surveys provide academic staff with evidence of the effectiveness of their teaching and identify areas for improvement. Individual Monquest reports are confidential, however, you can see the summary results of Monquest evaluations for 2006 at <http://www.adm.monash.edu.au/cheq/evaluations/monquest/profiles/index.html>

## **Unit staff - contact details**

### **Unit leader**

**Mr Michael Smith**

**Lecturer(s) :**

**Mr Michael Smith**

## Teaching and learning method

This unit will be delivered via two one hour lectures. The lecturer may go through specific examples, give demonstrations and present slides that contain theoretical concepts.

In tutorials students will discuss in-depth fundamental and interesting aspects about programming which will help them complete their practical work. The tutorials are particularly useful in helping students consolidate concepts and practise their problem solving skills.

Laboratories will be devoted to giving students hands-on experience in implementing a programming solution to a practical problem.

## Tutorial allocation

Students should register for tutorials/laboratories using Allocate+.

## Communication, participation and feedback

Monash aims to provide a learning environment in which students receive a range of ongoing feedback throughout their studies. You will receive feedback on your work and progress in this unit. This may take the form of group feedback, individual feedback, peer feedback, self-comparison, verbal and written feedback, discussions (on line and in class) as well as more formal feedback related to assignment marks and grades. You are encouraged to draw on a variety of feedback to enhance your learning.

It is essential that you take action immediately if you realise that you have a problem that is affecting your study. Semesters are short, so we can help you best if you let us know as soon as problems arise. Regardless of whether the problem is related directly to your progress in the unit, if it is likely to interfere with your progress you should discuss it with your lecturer or a Community Service counsellor as soon as possible.

## Unit Schedule

Week	Topic	Key dates
1	Introduction, Threads	Week 1 Labs - Diagnostic Quiz (Hurdle)
2	Architecture, TCP/IP, Sockets	
3	Protocol Design	
4	State Transitions, Half-Object Plus Protocol, Structured Data Formats	
5	General Security, Java Security	
6	HTTP, HTTP and Java	
7	Remote Procedure Call, Web Services	Assignment 1 due Wednesday, 27th August
8	CORBA	
9	RMI, Jeri	
10	Multiple Middleware, Jini	
11	JMS, JXTA	
Mid semester break		



CPE4001 Distributed programming in Java - Semester 2 , 2008

12	JavaSpaces	Assignment 2 due Sunday, 12th October
13	Revision	

## Unit Resources

### Prescribed text(s) and readings

None

### Recommended text(s) and readings

- An Introduction to Network Programming with Java [electronic resource] by Jan Graba
- Java Network Programming: A Complete *Guide to Networking, Streams, and Distributed Computing* by Merlin Hughes, Michael Shoffner, Derek Hamner
- A Programmers Guide to Jini by Jan Newmarch
- Java Programming with CORBA by Andreas Vogel and Keith Duddy
- Java.rmi : The Remote Method Invocation Guide by Esmond Pitt, Kathleen McNiff
- Core Java 2, Volume 2: Advanced Features by Cay S. Horstmann, Gary Cornell
- Big Java. 3rd Edition. Horstmann, C.S. (2008).

Further References:

- Arnold, K et al. The Jini (TM) Specification. Addison-Wesley, 1999
- Farley J and Farley J Java Distributed Computing, O'Reilly and Associates, 1998
- Freeman E et al. JavaSpaces(TM) Principles, Patterns and Practice. Addison-Wesley, 1999
- Lea, D Concurrent Programming in Java. Addison-Wesley, 1997
- Lewis G et al Programming with Java IDL, John Wiley, 1997 O
- Orfali et al The Essential Distributed Objects Survival Guide, John Wiley, 1996
- Orfali R and Harkey D Client/Server Programming with Java and CORBA, 2nd edition, John Wiley, 1998

### Required software and/or hardware

You will need access to:

- Java Version 6 (download from [java.sun.com](http://java.sun.com))
- Microsoft Access 2003

On-campus students may use this software which is installed in the computing labs. Information about computer use for students is available from the ITS Student Resource Guide in the Monash University Handbook.

### Equipment and consumables required or provided

Students may use the facilities available in the computing labs. Information about computer use for students is available from the ITS Student Resource Guide in the Monash University Handbook. You will need to allocate up to 8 hours per week for use of a computer, including time for newsgroups/discussion groups.

### Study resources

Study resources we will provide for your study are:

Study resources we will provide for your study are:

- Weekly detailed lecture notes outlining the learning objectives, discussion of the content, required readings and exercises;
- Weekly tutorial or laboratory tasks and exercises.
- Assignment specifications.
- This Unit Guide outlining the administrative information for the unit;
- The unit web site on MUSO, where resources outlined above will be made available.

## Library access

The Monash University Library site contains details about borrowing rights and catalogue searching. To learn more about the library and the various resources available, please go to <http://www.lib.monash.edu.au>. Be sure to obtain a copy of the Library Guide, and if necessary, the instructions for remote access from the library website.

## Monash University Studies Online (MUSO)

All unit and lecture materials are available through MUSO (Monash University Studies Online). Blackboard is the primary application used to deliver your unit resources. Some units will be piloted in Moodle. If your unit is piloted in Moodle, you will see a link from your Blackboard unit to Moodle (<http://moodle.monash.edu.au>) and can bookmark this link to access directly. In Moodle, from the Faculty of Information Technology category, click on the link for your unit.

You can access MUSO and Blackboard via the portal: <http://my.monash.edu.au>

Click on the Study and enrolment tab, then Blackboard under the MUSO learning systems.

In order for your Blackboard unit(s) to function correctly, your computer needs to be correctly configured.

For example:

- Blackboard supported browser
- Supported Java runtime environment

For more information, please visit: <http://www.monash.edu.au/muso/support/students/downloadables-student.html>

**You can contact the MUSO Support by: Phone: (+61 3) 9903 1268**

For further contact information including operational hours, please visit:  
<http://www.monash.edu.au/muso/support/students/contact.html>

Further information can be obtained from the MUSO support site:  
<http://www.monash.edu.au/muso/support/index.html>

## Assessment

### Unit assessment policy

To pass this unit, a student must obtain :

- 40% or more in the unit's examination and
- 40% or more in the unit's non-examination assessment  
and
- an overall unit mark of 50% or more

If a student does not achieve 40% or more in the unit examination or the unit non-examination assessment then a mark of no greater than 44-N will be recorded for the unit.

### Assignment tasks

#### • Assignment Task

**Title :** Diagnostic Quiz

**Description :**

In the tutorial labs during the first week of semester students will be required to undertake a quiz on assumed Java programming knowledge.

The quiz is designed to help students and provide feedback by highlighting gaps or weaknesses in the student's current knowledge of Java. It will therefore serve as a guide to any necessary private study needed to refresh, or learn new, Java concepts.

**Weighting :** 0% (Hurdle)

**Criteria for assessment :**

Although a compulsory hurdle, this task is worth no marks. Students can expect feedback but no grade or mark grade will be recorded.

**Due date :** Week 1 Labs

#### • Assignment Task

**Title :** Tutorial Participation

**Description :**

Students will be assessed on their contribution and participation in discussions in the tutorial lab classes during semester.

**Weighting :** 10%

**Criteria for assessment :**

Details will be explained during the labs in the first week of semester

**Due date :** Ongoing throughout semester

- **Assignment Task**

**Title :** Assignment 1

**Description :**

The design and implementation of a distributed system employing the principles and technologies introduced in the first half of the semester.

**Weighting :** 20%

**Criteria for assessment :**

The assignment will be graded on design and functionality as well as the student's performance at a compulsory interview and demonstration of the program following the assignment submission.

Further details of the assessment criteria will be provided in the assignment specification.

**Due date :** 12 midnight, Wednesday 27th August, 2008

- **Assignment Task**

**Title :** Assignment 2

**Description :**

The design and implementation of a distributed system employing the principles and technologies introduced in the second half of the semester.

**Weighting :** 20%

**Criteria for assessment :**

The assignment will be graded on design and functionality as well as the student's performance at a compulsory interview and demonstration of the program following the assignment submission.

Further details of the assessment criteria will be provided in the assignment specification.

**Due date :** 12 midnight, Sunday 12th October, 2008

## **Examinations**

- **Examination**

**Weighting :** 50%

**Length :** 3 hours

**Type ( open/closed book ) :** Closed book

## **Assignment submission**

The Diagnostic Quiz will be submitted in the tutorial lab in which it is undertaken.

Assignments 1 and 2 will be submitted via Blackboard.

Further details will be provided in the assignment specification.

## University and Faculty policy on assessment

### Due dates and extensions

The due dates for the submission of assignments are given in the previous section. Please make every effort to submit work by the due dates. It is your responsibility to structure your study program around assignment deadlines, family, work and other commitments. Factors such as normal work pressures, vacations, etc. are seldom regarded as appropriate reasons for granting extensions. Students are advised to NOT assume that granting of an extension is a matter of course.

Requests for extensions must be made to the unit lecturer at your campus at least two days before the due date. You will be asked to forward original medical certificates in cases of illness, and may be asked to provide other forms of documentation where necessary. A copy of the email or other written communication of an extension must be attached to the assignment submission.

### Late assignment

Assignments received after the due date will be subject to a penalty of 10% per day, including weekends. Assignments received later than one week (seven days) after the due date will not be accepted.

This policy is strict because comments or guidance will be given on assignments as they are returned, and sample solutions may also be published and distributed, after assignment marking or with the returned assignment.

### Return dates

Students can expect assignments to be returned within two weeks of the submission date or after receipt, whichever is later.

Assessment for the unit as a whole is in accordance with the provisions of the Monash University Education Policy at <http://www.policy.monash.edu/policy-bank/academic/education/assessment/>

We will aim to have assignment results made available to you within two weeks after assignment receipt.

### Plagiarism, cheating and collusion

Plagiarism and cheating are regarded as very serious offences. In cases where cheating has been confirmed, students have been severely penalised, from losing all marks for an assignment, to facing disciplinary action at the Faculty level. While we would wish that all our students adhere to sound ethical conduct and honesty, I will ask you to acquaint yourself with Student Rights and Responsibilities (<http://www.infotech.monash.edu.au/about/committees-groups/facboard/policies/studrights.html>) and the Faculty regulations that apply to students detected cheating as these will be applied in all detected cases.

In this University, cheating means seeking to obtain an unfair advantage in any examination or any other written or practical work to be submitted or completed by a student for assessment. It includes the use, or attempted use, of any means to gain an unfair advantage for any assessable work in the unit, where the means is contrary to the instructions for such work.

When you submit an individual assessment item, such as a program, a report, an essay, assignment or other piece of work, under your name you are understood to be stating that this is your own work. If a submission is identical with, or similar to, someone else's work, an assumption of cheating may arise. If you are planning on working with another student, it is acceptable to undertake research together, and discuss problems, but it is not acceptable to jointly develop or share solutions unless this is specified by your lecturer.

Intentionally providing students with your solutions to assignments is classified as "assisting to cheat" and students who do this may be subject to disciplinary action. You should take reasonable care that your solution is not accidentally or deliberately obtained by other students. For example, do not leave copies of your work in progress on the hard drives of shared computers, and do not show your work to other students. If you believe this may have happened, please be sure to contact your lecturer as soon as possible.

Cheating also includes taking into an examination any material contrary to the regulations, including any bilingual dictionary, whether or not with the intention of using it to obtain an advantage.

Plagiarism involves the false representation of another person's ideas, or findings, as your own by either copying material or paraphrasing without citing sources. It is both professional and ethical to reference clearly the ideas and information that you have used from another writer. If the source is not identified, then you have plagiarised work of the other author. Plagiarism is a form of dishonesty that is insulting to the reader and grossly unfair to your student colleagues.

## **Register of counselling about plagiarism**

The university requires faculties to keep a simple and confidential register to record counselling to students about plagiarism (e.g. warnings). The register is accessible to Associate Deans Teaching (or nominees) and, where requested, students concerned have access to their own details in the register. The register is to serve as a record of counselling about the nature of plagiarism, not as a record of allegations; and no provision of appeals in relation to the register is necessary or applicable.

## **Non-discriminatory language**

The Faculty of Information Technology is committed to the use of non-discriminatory language in all forms of communication. Discriminatory language is that which refers in abusive terms to gender, race, age, sexual orientation, citizenship or nationality, ethnic or language background, physical or mental ability, or political or religious views, or which stereotypes groups in an adverse manner. This is not meant to preclude or inhibit legitimate academic debate on any issue; however, the language used in such debate should be non-discriminatory and sensitive to these matters. It is important to avoid the use of discriminatory language in your communications and written work. The most common form of discriminatory language in academic work tends to be in the area of gender inclusiveness. You are, therefore, requested to check for this and to ensure your work and communications are non-discriminatory in all respects.

## **Students with disabilities**

Students with disabilities that may disadvantage them in assessment should seek advice from one of the following before completing assessment tasks and examinations:

- Faculty of Information Technology Student Service staff, and / or
- your Unit Coordinator, or
- [Disabilities Liaison Unit](#)

## **Deferred assessment and special consideration**

Deferred assessment (not to be confused with an extension for submission of an assignment) may be granted in cases of extenuating personal circumstances such as serious personal illness or bereavement. Information and forms for Special Consideration and deferred assessment applications are available at <http://www.monash.edu.au/exams/special-consideration.html>. Contact the Faculty's Student Services staff at your campus for further information and advice.