



**MONASH** University  
Information Technology

**FIT5170**  
**Programming for distributed, parallel and mobile systems**

**Unit Guide**

**Semester 1, 2011**

The information contained in this unit guide is correct at time of publication. The University has the right to change any of the elements contained in this document at any time.

*Last updated: 27 Feb 2011*

# Table of Contents

<a href="#"><u>FIT5170 Programming for distributed, parallel and mobile systems - Semester 1, 2011</u></a> .....	1
<a href="#"><u>Mode of Delivery</u></a> .....	1
<a href="#"><u>Contact Hours</u></a> .....	1
<a href="#"><u>Workload</u></a> .....	1
<a href="#"><u>Unit Relationships</u></a> .....	1
<a href="#"><u>Prerequisites</u></a> .....	1
<a href="#"><u>Chief Examiner</u></a> .....	1
<a href="#"><u>Campus Lecturer</u></a> .....	1
<a href="#"><u>Caulfield</u></a> .....	1
<a href="#"><u>Learning Objectives</u></a> .....	2
<a href="#"><u>Graduate Attributes</u></a> .....	2
<a href="#"><u>Assessment Summary</u></a> .....	2
<a href="#"><u>Teaching Approach</u></a> .....	2
<a href="#"><u>Feedback</u></a> .....	3
<a href="#"><u>Our feedback to You</u></a> .....	3
<a href="#"><u>Your feedback to Us</u></a> .....	3
<a href="#"><u>Previous Student Evaluations of this unit</u></a> .....	3
<a href="#"><u>Required Resources</u></a> .....	3
<a href="#"><u>Unit Schedule</u></a> .....	3
<a href="#"><u>Assessment Policy</u></a> .....	4
<a href="#"><u>Assessment Tasks</u></a> .....	4
<a href="#"><u>Participation</u></a> .....	4
<a href="#"><u>Examinations</u></a> .....	6
<a href="#"><u>Examination 1</u></a> .....	6
<a href="#"><u>Assignment submission</u></a> .....	6
<a href="#"><u>Extensions and penalties</u></a> .....	6
<a href="#"><u>Returning assignments</u></a> .....	6
<a href="#"><u>Policies</u></a> .....	6
<a href="#"><u>Student services</u></a> .....	7

# **FIT5170 Programming for distributed, parallel and mobile systems - Semester 1, 2011**

This unit focuses on the design and programming techniques essential for developing distributed software systems and applications - with Java as the teaching language. The unit presents concurrent programming primitives and concepts for distributed systems. The unit also focuses on application of concurrent techniques in distributed system designs. Programming and implementation issues and techniques of distributed applications are studied. Enabling techniques for building distributed systems are analyzed and evaluated. Distributed Software Patterns are presented. The unit also includes case studies of distributed programming paradigms and their applications (e.g. JINI, JavaSpaces).

## **Mode of Delivery**

Caulfield (Day)

## **Contact Hours**

2 hrs lectures/wk, 2 hrs laboratories/wk

## **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**

Recommended knowledge: Some exposure to multithreading. Knowledge of all Java language constructs such as loops, conditionals, methods, classes, inheritance and core Java packages. Use of O/O models such as UML diagrams.

### **Chief Examiner**

Chris Ling

### **Campus Lecturer**

#### **Caulfield**

Michael Smith

Contact hours: To be advised

## Learning Objectives

At the completion of this unit students will:

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

## Graduate Attributes

Monash prepares its graduates to be:

1. responsible and effective global citizens who:

- a. engage in an internationalised world
- b. exhibit cross-cultural competence
- c. demonstrate ethical values

critical and creative scholars who:

- a. produce innovative solutions to problems
- b. apply research skills to a range of challenges
- c. communicate perceptively and effectively

## Assessment Summary

Examination (3 hours): 50%, In-semester assessment: 50%

<b>Assessment Task</b>	<b>Value</b>	<b>Due Date</b>
Diagnostic Quiz	0% (Hurdle)	Week 1 Labs
Assignment 1	25%	Week 6 - date to be advised in the assignment specification.
Assignment 2	25%	Week 12 - date to be advised in assignment specification.
Examination 1	50%	To be advised

## Teaching Approach

This teaching and learning approach provides facilitated learning, practical exploration and peer learning.

## Feedback

### Our feedback to You

Types of feedback you can expect to receive in this unit are:

- Graded assignments with comments
- Interviews

### Your feedback to Us

Monash is committed to excellence in education and regularly seeks feedback from students, employers and staff. One of the key formal ways students have to provide feedback is through SETU, Student Evaluation of Teacher and Unit. The University's student evaluation policy requires that every unit is evaluated each year. Students are strongly encouraged to complete the surveys. The feedback is anonymous and provides the Faculty with evidence of aspects that students are satisfied and areas for improvement.

For more information on Monash's educational strategy, and on student evaluations, see:

<http://www.monash.edu.au/about/monash-directions/directions.html>

<http://www.policy.monash.edu/policy-bank/academic/education/quality/student-evaluation-policy.html>

### Previous Student Evaluations of this unit

If you wish to view how previous students rated this unit, please go to

<https://emuapps.monash.edu.au/unitevaluations/index.jsp>

### Required Resources

- Java Version 6 (download from java.sun.com)
- Microsoft Access 2007

### Unit Schedule

Week	Date*	Activities	Assessment
0	21/02/11		No formal assessment or activities are undertaken in week 0
1	28/02/11	Introduction, Threads	Week 1 Labs - Diagnostic Quiz (Hurdle)
2	07/03/11	Architecture, TCP/IP, Sockets	
3	14/03/11	Protocol Design	
4	21/03/11	State Transitions, Half-Object Plus Protocol, Structured Data Formats	
5	28/03/11	General Security, Java Security	
6	04/04/11	HTTP, HTTP and Java	Assignment 1 due
7	11/04/11	RMI	

8	18/04/11	CORBA	
Mid semester break			
9	02/05/11	Web Services	
10	09/05/11	Mobile Java	
11	16/05/11	Jini	
12	23/05/11	JavaSpaces, JMS, JXTA	Assignment 2 due
	30/05/11	SWOT VAC	No formal assessment is undertaken in SWOT VAC

\*Please note that these dates may only apply to Australian campuses of Monash University. Off-shore students need to check the dates with their unit leader.

## Assessment Policy

To pass a unit which includes an examination as part of the assessment a student must obtain:

- 40% or more in the unit's examination, and
- 40% or more in the unit's total 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 total assessment, and the total mark for the unit is greater than 50% then a mark of no greater than 49-N will be recorded for the unit

## Assessment Tasks

### Participation

- **Assessment task 1**

**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:**

Demonstrated ability to implement complete and appropriate solutions to prescribed problems.

**Due date:**

Week 1 Labs

• **Assessment task 2**

**Title:**

Assignment 1

**Description:**

The design and implementation of a distributed system employing the principles and technologies introduced in the early to mid part of the semester.

**Weighting:**

25%

**Criteria for assessment:**

This is an individual assignment and is to be entirely your own work.

Assessment for this assignment is by interview. You will be asked to demonstrate your system at an interview in the week following the submission date. At the interview you can also expect to be asked to explain your system, your code, your design, discuss design decisions and alternatives and modify your code/system as required. Marks will not be awarded for any section of code or functionality that a student cannot explain satisfactorily. (The marker may delete excessive comments in code before a student is asked to explain that code).

Interview times will be arranged in the tutorial labs immediately preceding the submission deadline. **It is your responsibility to attend the lab and obtain an interview time.** *Students who do not attend an interview will receive 0 marks for the assignment.*

Further detailed assessment criteria will be available with the assignment specification

**Due date:**

Week 6 - date to be advised in the assignment specification.

**Remarks:**

The details of the task and other requirements will be outlined in the assignment specification

• **Assessment task 3**

**Title:**

Assignment 2

**Description:**

The design and implementation of a distributed system employing the principles and technologies introduced in the mid to later part of the semester.

**Weighting:**

25%

**Criteria for assessment:**

This is an individual assignment and is to be entirely your own work.

Assessment for this assignment is by interview. You will be asked to demonstrate your system at an interview in the week following the submission date. At the interview you can also expect to be asked to explain your system, your code, your design, discuss design decisions and alternatives and modify your code/system as required. Marks will not be awarded for any section of code or functionality that a student cannot explain satisfactorily. (The marker may delete excessive comments in code before a student is asked to explain that code).

Interview times will be arranged in the tutorial labs immediately preceding the submission deadline. **It is your responsibility to attend the lab and obtain an interview time.** *Students who do not attend an interview will receive 0 marks for the assignment.*

Further detailed assessment criteria will be available with the assignment specification

**Due date:**

Week 12 - date to be advised in assignment specification.

**Remarks:**

The details of the task and other requirements will be outlined in the assignment specification

## Examinations

- **Examination 1**

**Weighting:**

50%

**Length:**

3 hours

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

Closed book

**Electronic devices allowed in the exam:**

None

## Assignment submission

Assignment coversheets are available via "Student Forms" on the Faculty website:

<http://www.infotech.monash.edu.au/resources/student/forms/>

You MUST submit a completed coversheet with all assignments, ensuring that the plagiarism declaration section is signed.

## Extensions and penalties

Submission must be made by the due date otherwise penalties will be enforced.

You must negotiate any extensions formally with your campus unit leader via the in-semester special consideration process:

<http://www.infotech.monash.edu.au/resources/student/equity/special-consideration.html>.

## Returning assignments

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

## Policies

Monash has educational policies, procedures and guidelines, which are designed to ensure that staff and students are aware of the University's academic standards, and to provide advice on how they might uphold them. You can find Monash's Education Policies at:

<http://policy.monash.edu.au/policy-bank/academic/education/index.html>

Key educational policies include:

- Plagiarism

- (<http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-policy.html>)
- Assessment  
(<http://www.policy.monash.edu/policy-bank/academic/education/assessment/assessment-in-coursework-p>)
- Special Consideration  
(<http://www.policy.monash.edu/policy-bank/academic/education/assessment/special-consideration-policy.h>)
- Grading Scale  
(<http://www.policy.monash.edu/policy-bank/academic/education/assessment/grading-scale-policy.html>)
- Discipline: Student Policy  
(<http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-discipline-policy.html>)
- Academic Calendar and Semesters (<http://www.monash.edu.au/students/key-dates/>);
- Orientation and Transition (<http://www.infotech.monash.edu.au/resources/student/orientation/>);  
and
- Academic and Administrative Complaints and Grievances Policy  
(<http://www.policy.monash.edu/policy-bank/academic/education/management/complaints-grievance-policy>)

## Student services

The University provides many different kinds of support services for you. Contact your tutor if you need advice and see the range of services available at [www.monash.edu.au/students](http://www.monash.edu.au/students) The Monash University Library provides a range of services and resources that enable you to save time and be more effective in your learning and research. Go to <http://www.lib.monash.edu.au> or the library tab in my.monash portal for more information. Students who have a disability or medical condition are welcome to contact the Disability Liaison Unit to discuss academic support services. Disability Liaison Officers (DLOs) visit all Victorian campuses on a regular basis

- Website: <http://adm.monash.edu/sss/equity-diversity/disability-liaison/index.html>;
- Telephone: 03 9905 5704 to book an appointment with a DLO;
- Email: [dlu@monash.edu](mailto:dlu@monash.edu)
- Drop In: Equity and Diversity Centre, Level 1 Gallery Building (Building 55), Monash University, Clayton Campus.

## Recommended Reading

- 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).