

### **INFO1103: INTRODUCTION TO PROGRAMMING**

Semester 1, 2016 | 6 Credit Points | Mode: Normal-Day

Sessions Valid: Semester 1, Semester 2 Coordinator(s): John Stavrakakis

WARNING: This unit version is currently being edited and is subject to change!

### 1. INTRODUCTION

Programming in a legible, maintainable, reusable way is essential to solve complex problems in the pervasive computing environments. This unit will equip students with foundation of programming concepts that are common to widely used programming languages. The "fundamentals-first & objects-later" strategy is used to progressively guide this introductory unit from necessary and important building blocks of programming to the object-oriented approach. Java, one of the most popular programming languages, is used in this unit. It provides interdisciplinary approaches, applications and examples to support students from broad backgrounds such as science, engineering, and mathematics.

### 2. LEARNING OUTCOMES

Learning outcomes are the key abilities and knowledge that will be assessed in this unit. See assessment summary table below for details of which outcomes are assessed where. Outcomes are listed according to the course goals that they support.

#### Engineering/IT Specialisation (Level 1)

- 1. Ability to read, understand and trace Java code
- 2. Basic experience of testing and debugging for basic Java programs
- 3. Understanding of the concepts of inheritance, polymorphism and interfaces.
- 4. Understanding of basic Java programming
- 5. Understanding of the fundamental concepts of object oriented programming
- 6. Ability to understand, modify and add functionality to Java programs
- 7. Ability to convert simple pseudo-code into Java code
- 8. Experience using the Java collections framework

For further details of course goals related to these learning outcomes, see online unit outline at <a href="http://cusp.eng.usyd.edu.au/students/view-unit-page/alpha/INFO1103">http://cusp.eng.usyd.edu.au/students/view-unit-page/alpha/INFO1103</a>.

## 3. ASSESSMENT TASKS

# ASSESSMENT SUMMARY

Assessment name	Team-based?	Weight	Due	<b>Outcomes Assessed</b>
Online Challenges	No	5%	Multiple Weeks	1, 2, 3, 4, 5, 6, 7, 8
*Assignment 1	No	10%	Week 5 (Monday, 9 am)	2, 4, 6, 7
*Quiz 1	No	7.5%	Week 8	1, 2, 4, 5
*Assignment 2	No	10%	Week 9 (Monday, 9 am)	1, 2, 4, 5, 6, 7
*Quiz 2	No	7.5%	Week 12	1, 2, 4, 6, 7
*Assignment 3	No	10%	Week 13 (Monday, 9 am)	1, 2, 3, 4, 5, 6, 7, 8
Final Exam	No	50%	Exam Period	1, 2, 3, 4, 5, 6, 7, 8

#### ASSESSMENT DESCRIPTION

\* indicates an assessment task which must be repeated if a student misses it due to special consideration. A repeated assessment's questions/tasks may vary from the original.

Homework (Online challenges) - 5%

- Challenges are released regularly and have varying submission dates.
- Late enrolments (up to week 4) will be able to submit prior weeks of a similar task

Quiz 1 & 2 - 15%

- 40 minutes: A pen and paper exercise to test both knowledge and skills of course materials in the semester thus far.

Assignments - 30%

- Demonstrating design and programming ability. No late submissions except by Special Consideration.

Final Exam - 50%

- 2 hours: A pen and paper formal examination.

Late assignment submission is 20% per day.

Non-attendance for in lab assessment tasks, such as manual marking to exercises will award zero marks, unless special consideration is granted.

#### ASSESSMENT FEEDBACK

The tutor will be grading the assignment and will provide the information to the student in the software/websites available.

#### **ASSESSMENT GRADING**

Final grades in this unit are awarded at levels of HD for High Distinction, DI (previously D) for Distinction, CR for Credit, PS (previously P) for Pass and FA (previously F) for Fail as defined by University of Sydney Assessment Policy. Details of the Assessment Policy are available on the Policies website at <a href="http://sydney.edu.au/policies">http://sydney.edu.au/policies</a>. Standards for grades in individual assessment tasks and the summative method for obtaining a final mark in the unit will be set out in a marking guide supplied by the unit coordinator.

It is a policy of the School of Information Technologies that in order to pass this unit, a student must achieve at least 40% in the written examination. For subjects without a final exam, the 40% minimum requirement applies to the corresponding major assessment component specified by the lecturer. A student must also achieve an overall final mark of 50 or more. Any student not meeting these requirements may be given a maximum final mark of no more than 45 regardless of their average.

#### IMPORTANT: POLICY RELATING TO ACADEMIC DISHONESTY AND PLAGIARISM.

All students must submit a cover sheet for all assessment work that declares that the work is original and not plagiarised from the work of others

In assessing a piece of submitted work, the School of IT may reproduce it entirely, may provide a copy to another member of faculty, and/or to an external plagiarism checking service or in-house computer program and may also maintain a copy of the assignment for future checking purposes and/or allow an external service to do so.

See Policies section below for other policies relating to assessment and progression.

#### 4. ATTRIBUTES DEVELOPED

Attributes listed here represent the course goals designated for this unit. The list below describes how these attributes are developed through practice in the unit. See Learning Outcomes and Assessment sections above for details of how these attributes are assessed.

Attribute	Method
Design (Level 2)	Algorithm-based approach to problem solving
Engineering/IT Specialisation (Level 1)	Basic programming and use of algorithms
Information Seeking (Level 1)	Problem solving will require locating and using information effectively and efficiently.
Communication (Level 1)	Lab work will encourage collaboration, team work and communication.

For further details of course goals and professional attribute standards, see the online version of this outline at <a href="http://cusp.eng.usyd.edu.au/students/view-unit-page/alpha/INFO1103">http://cusp.eng.usyd.edu.au/students/view-unit-page/alpha/INFO1103</a>.

# **5. STUDY COMMITMENT**

About the seminar:

- Attendance is not compulsory
- It will not contain new content needed to complete this course
- Seminar has value to those who need more help with reviewing material and programming exercises

Activity	Hours per Week	Sessions per Week	Weeks per Semester
Lecture	1.00	2	13
Laboratory	2.00	1	13
Independent Study	8.00		13
Seminar	1.00	1	12

Standard unit of study workload at this university should be from 1.5 to 2 hours per credit point which means 9-12 hours for a normal 6 credit point unit of study. For units that are based on research or practical experience, hours may vary. For lecture and tutorial timetable, see University timetable site at: web.timetable.usyd.edu.au/calendar.jsp

#### 6. TEACHING STAFF AND CONTACT DETAILS

# COORDINATOR(S)

NameRoomPhoneEmailContact noteDr Stavrakakis, Johnjohn.stavrakakis@sydney.edu.au

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_

**LECTURERS** 

Name Room Phone Email Contact note

Dr Stavrakakis, John john.stavrakakis@sydney.edu.au

**TUTORS** 

Zhizhou (Joe) YIN

Mansour Khelghatdoust

Waiho Wong

The Trung (James) Nguyen

Scott Maxwell

Dean Codemo

Xavier Holt

Eric Liu

Alan Robertson

Natalie Tridgell

Yu Zhao

Farahnaz Yekeh

Elie Moreau

### 7. RESOURCES

# PRESCRIBED TEXTBOOK(S)

Walter Savitch, Java: An Introduction to Problem Solving and Programming, 7th Edition (7). USA, Pearson Higher Ed USA, 2014. 9781292018331.

### **RECOMMENDED REFERENCES**

Cay Horstmann, Big Java (4). John Wiley & Sons, Inc, 2009. 978-0-470-50948-7.

Robert Sedgewick and Kevin Wayne, Introduction to Programming in Java (1). Addison Wesley, 2007. 0321498054.

Cay Horstmann, Java Concepts (6). John Wiley & Sons, Inc, 2009. 978-0-470-50947-0.

### **NOTE ON RESOURCES**

There is a prescribed text for this unit. Others are highly recommended.

The text is used to provide extra support for reading and practice and its exercises are an excellent way to study for this unit.

Each chapter also has a graphics supplement, which is not required material but it is intended that they will provide a suitable challenge for students looking to stretch themselves.

# **8. ENROLMENT REQUIREMENTS**

# **ASSUMED KNOWLEDGE**

None.

#### **PREREQUISITES**

None.

# 9. POLICIES

# IMPORTANT: School policy relating to Academic Dishonesty and Plagiarism.

In assessing a piece of submitted work, the School of IT may reproduce it entirely, may provide a copy to another member of faculty, and/or to an external plagiarism checking service or in-house computer program and may also maintain a copy of the assignment for future checking purposes and/or allow an external service to do so.

### Other policies

See the policies page of the faculty website at http://sydney.edu.au/engineering/student-policies/ for information regarding university policies and local provisions and procedures within the Faculty of Engineering and Information Technologies.

# 10. WEEKLY SCHEDULE

Note that the "Weeks" referred to in this Schedule are those of the official university semester calendar <a href="https://web.timetable.usyd.edu.au/calendar.jsp">https://web.timetable.usyd.edu.au/calendar.jsp</a>

Week	Topics/Activities
Week 1	Lecture: Introduction to the unit, fundamental concepts, first program
	Lab: Programming basics
Week 2	Lecture: Numerical, Boolean and String variables; expressions
	Lab: Variables and Types
Week 3	Lecture: Control Flow 1: Branching (IF and WHILE)
	Lab: Branching and Loops
Week 4	Lecture: Control Flow 2: iteration (FOR), Arrays
	Lab: Loops and the Design Process
Week 5	Lecture: Methods
	Lab: Arrays and Objects
	Assessment Due: *Assignment 1
Week 6	Lecture: Exceptions, File input/output
	Lab: Methods and Exceptions
Week 7	Lecture: Classes and Objects
	Lab: Files and Classes
Week 8	Lecture: Classes and methods, testing
	Lab: More on classes
	Assessment Due: *Quiz 1
Week 9	Lecture: Design problems with Classes and Methods
	Lab: Testing and Multi-class Programs
	Assessment Due: *Assignment 2
Week 10	Lecture: Programming idioms, inheritance
	Lab: Using classes
Week 11	Lecture: Collections, Recursion
	Lab: Idioms and Inheritance
Week 12	Lecture: Inheritance, Multidimensional arrays
	Assessment Due: *Quiz 2
Week 13	Lecture: Revision and Examination overview
	Lab: Revision
	Assessment Due: *Assignment 3
Exam Period	Assessment Due: Final Exam