

|                           |  |               |       |                   |     |
|---------------------------|--|---------------|-------|-------------------|-----|
| <b>Course Name</b>        | <b>PROGRAMMING IN JAVA 2</b>               |               |       |                   |     |
| <b>Previous Name</b>      | Java Programming Lab                       |               |       |                   |     |
| <b>Course ID</b>          | 012524                                     |               |       |                   |     |
| <b>Area &amp; Cat No.</b> | COMP 1009                                  |               |       |                   |     |
| <b>School</b>             | CIS  | <b>Campus</b> | SAIBT |                   |     |
| <b>Contact hours</b>      | Tutorial (12 x 4.25)<br>Practical (12 x 2) | <b>Mode</b>   | INT   | <b>Final Exam</b> | Yes |

### Prerequisites

Programming in Java 1

### Literacy Requirements

The literacy requirements are consistent with the English language entry requirements for the Program.

### Aim

To introduce concepts and techniques for object-oriented programming and for program development..

### Objectives

On completion of the course, students should be able to:

- apply the principles of object-orientated programming;
- write object-oriented code in Java;
- Develop class hierarchies using Inheritance
- Use object-orientated class libraries for Exceptions and Data streams
- Understand and build programs using recursive algorithms
- Build and use Linked Lists, Stacks and Queues
- Use Java collection classes
- Apply formal program design techniques to design and develop simple applications
- Use structured techniques for program testing and debugging

### Syllabus

Inheritance. Polymorphism, Overriding of inherited methods .. Abstract Classes and Interfaces. Reuse of classes through inheritance and composition. Abstraction and encapsulation. Exception handling (defining exception classes, throw, throws, and try-catch-final). Exception hierarchies. The Java I/O system (file types, text I/O, binary I/O, object serialization). Recursion. Sorting of objects using the Comparable Interface, Linked lists, stacks and queues. Java Collection classes, Templates. Basic UML notation (class diagrams, associations, state diagrams). Program design (identifying classes and methods, UML diagrams, code conversion). Program testing (white box: program flow and boundary, black box), program debugging. Java program documentation.

### Teaching and Learning Arrangements

This course will be delivered using the following means:

- Tutorials, to provide the theoretical basis of the subject and provide an opportunity for students to discuss relevant topics
- Practicals, to consolidate a student's understanding of programming techniques.

### Assessment

|              |     |
|--------------|-----|
| Assignment 1 | 10% |
| Test         | 10% |
| Assignment 2 | 20% |
| Examination  | 60% |

### Textbook

Savitch, W (2005) *Absolute Java 2<sup>nd</sup> ed*, Addison Wesley

### References

- Oestereich, B (2000) *Developing Software with UML, Object-oriented Analysis and Design in Practice* Addison Wesley
- Pressman, RS (2000), *Software-Engineering – A practitioner's approach* McGraw-Hill
- Louden, K (1993) *Programming Languages, Principles and Practice* PWS Publishing Company

### University of South Australia Course Coordinator

Phillip Lock