IT2203: PROGRAMMING I

(Compulsory)

1. OUTLINE OF SYLLABUS

Topic	Minimum number of hours
• 1 - Introduction to Programming	02
• 2 - Interacting with Java Programs	03*
• 3 - Fundamentals of Java Programming	15*
• 4 - Flow control	06*
• 5 - Program Design Techniques	07
• 6 - Object Oriented Concepts and Techniques	15*
• 7 - String Handling	04*
• 8 - Error Handling in Java	04*
• 9 - Streams in Java	04*
Total for the subject	**60**

* *Students are expected to have a total of 35 hours of practical and tutorials to strengthen their knowledge of these sections*

Learning Outcome

After successfully completing this module students will be able to design and develop programs for specified tasks using Java as an Object Oriented Programming Language.

2. DETAILED SYLLABUS

1. Introduction to Programming (02hrs)

Instructional Objectives
- Define what programming is
- List the steps of writing a program
- Describe advantages of using Java

List the difference between Java Runtime Environment (JRE) and Java 2 Platform, Standard Edition (J2SE)

1.1. What is programming? [Ref 1: pg 12 -19]

1.2. The required steps of writing a program [Ref 1: pg 12 - 27]
1.3. What is Java? [Ref 1: pg 1 - 4]
1.4. Java Virtual Machine (JVM) and Java Runtime Environment (JRE) [Ref 1: pg 4 - 6]
1.5. Advantages of using Java [Ref 1: pg 21]

2. Interacting with Java Programs (03hrs)

Instructional Objectives
- Install and set up the Java programming environment
- Explain the compiling and interpretation processes
- Compile and execute a Java program

2.1. Introduction to Java Development Kit (JDK) [Ref 1: pg 5 - 6]
2.2. Installing Java [Ref 1: pg 6 - 8]
2.3. Creating a source file [Ref 1: pg 24 - 26]
2.4. Compiling and running a source file [Ref 1: pg 8 - 12]

3. Fundamentals of Java Programming (15hrs)

Instructional Objectives
- Identify the structure of a Java program
- Identify the appropriate data types for representing different data
- Apply different operators to java Programs.
- Explain Java Programs by using different types of arrays
- List the operators according to their precedence levels.

3.1. Structure of a Java program [Ref 1: pg 19 - 24]
3.2. Statements [Ref 1: pg 19]
3.3. Comments in Java
  3.3.1. Standard comments [Ref 1: pg 81 - 82]
  3.3.2. Document comments [Ref 1: pg 82 - 83]
3.4. Literals [Ref 1: pg 31,33,36,153]
3.5. Variables and data types
  3.4.1 What is a variable? [Ref 1: pg 4 - 6]
  3.4.2. Declaring variables [Ref 1: pg 29 - 31]
  3.4.3. Conventions in variable naming [Ref 1: pg 4 - 6]
  3.4.4 Data types [Ref 1: pg 31 – 37,60 – 61, 79 - 80]
  3.4.5 Constants and enumerations [Ref 1: pg 37, 77 - 79]
3.6. Arrays
  3.5.1. One dimensional and array of arrays [Ref 1: pg 135 - 152]
    3.5.1.1. Declaring array variables
    3.5.1.2. Creating array objects
    3.5.1.3. Accessing array elements
    3.5.1.4. Changing array elements
    3.5.1.5. Array of arrays
3.7. Operators
  3.7.1. Arithmetic operators: + , - , * , / , % , ++ , -- [Ref 1: pg 38 - 52]
  3.7.2. Logical operators: && , || , !,&.^ [Ref 1: pg - 95 - 99]
  3.7.3. Bitwise operators: & , | , ^ , << , >> , >>> , ~ [Ref 1: pg 63 - 77]
  3.7.4. Relational operators: = = , != , < , > , <= , >= [Ref 1: pg 86 - 87]
  3.7.5. Conditional operator (Short hand operator for if-then-else)(?,:) and other operators [Ref 1: pg 53, 100 -101]
  3.7.6. Operator precedence [Ref 1: pg. 80 - 81]

4. Flow Control (6hrs)

Instructional Objectives
- Design an algorithm to solve a task by combining the various flow control statements in appropriate ways
- Describe the use of different flow control statements
- Compare the differences of flow control statements
- Note down Java programmes to solve problems using Flow Controls.

4.1. Flow control statements in Java
- 4.1.1. Selective statements [Ref 1: pg 85 - 108]
  - 4.1.1.1. Selection statements (if, if.... else, switch,)
- 4.1.2. Iterative statements [Ref 1: pg 111 - 121]
  - 4.1.2.1. For loop
  - 4.1.2.2. While loop
  - 4.1.2.3. Do-while
  - 4.1.2.4. Nested loops
- 4.1.3. Jump statements [Ref 1: pg 122 - 128]
  - 4.1.3.1. Break
  - 4.1.3.2. Continue
  - 4.1.3.3. Return
  - 4.1.3.4. Labelled loops

5. Program Design Techniques (07hrs)

Instructional Objectives
- List important steps of software development life cycle
- List different tools for program design
- List the evolution of software design
- Compare the tools for algorithm specification
- Note down data and procedures
- Describe program structure and modular design.

5.1. An overview of program design [Ref 2: pg. 1–10]
- 5.1.1. Importance of professional programming
- 5.1.2. Tools for program design
- 5.1.3. Evolution of software design
- 5.1.4. Tools for algorithm specification and comparison between them

5.2. An overview of data and procedures [Ref 2: pg. 11-31]
- 5.2.1. Data names
- 5.2.2. Explicit vs implicit definition of data
- 5.2.3. Data types and data usage
- 5.2.4. Data structures
- 5.2.5. Data scope
- 5.2.6. Simple procedures

5.3. An Overview of Program Structure, Modular design [Ref 2: pg. 46–56]
- 5.3.1. Top down decomposition
- 5.3.2. Principles for decomposition
- 5.3.3. Program structure
- 5.3.4. Communication between procedures
- 5.3.5. Sub routine and functions

6. Object Oriented Concepts and Techniques (15hrs)

Instructional Objectives
- Explain the concepts of object orientation.
- Design and develop object oriented Java programs

6.1. Definition of a class [Ref 1: pg 200]
6.2. Creating objects [Ref 1: pg 217 - 220]
6.3. Defining methods [Ref 1: pg 202 - 217]
6.4. Parameter passing: passing arguments to methods [Ref 1: pg 206 - 208]
6.5. Constructor methods: overloading constructors [Ref 1: pg 215 ,223 - 225]
6.6. *this* and *super* keywords [Ref 1: pg 210 – 211, 276 - 277]
6.7. Encapsulation (information hiding) [Ref 1: pg 246 – 250,200 – 202 ,209, 287, 737,]
  6.7.1. Access modifiers: controlling access to a class, method, or variable (public, protected, private, default)
  6.7.2. Other modifiers: static modifier, abstract modifier, final modifier, synchronised modifier
6.8. Inheritance [Ref 1: pg 270 – 279, 287 - 288]
  6.8.1. Inheritance basics
  6.8.2. Overriding methods
  6.8.3. Abstract classes
  6.8.4. Reusability
6.9. Polymorphism: overloading, overriding [Ref 1: pg 210 – 211, 276 - 277]
6.10 Recursions [Ref 1: pg 233 - 236]

7. String Handling (04hrs)

Instructional Objectives
- Identify the usage of Strings in programs
- Use various important String manipulation methods
- Use StringBuffer and StringTokenizer classes

7.1. Why Strings in a program? [Ref 1: pg 152 - 155]
7.2. String manipulation methods [Ref 1: pg 157 – 184]
7.3. StringBuffer class [Ref 1: pg 184 - 194]

8. Error Handling in Java (04hrs)

Instructional Objectives
- Handle errors in a program
- Write robust programs
- Process exceptions in Java programs

8.1. Why error handling is important? [Ref 1: pg 339]
8.2. Java way: Errors and Exceptions [Ref 1: pg 340 - 344]
8.3. Exception fundamentals: protecting code and catching exceptions try catch clause, finally clause [Ref 1: pg 345 - 358]
8.4. Handling exceptions [Ref 1: pg 364]
8.5. Throwing and rethrowing exceptions: throws clause [Ref 1: pg 344, 359 - 361]
8.6. Creating your own exception subclasses [Ref 1: pg362 – 368]

9. Streams in Java (04hrs)

Instructional Objectives

- Write programs that use input and output streams

9.1. Input and output [Ref 1: pg 371 – 398]

- 9.1.1. Streams and new I/O capabilities
- 9.1.2. Understanding streams
- 9.1.3. The classes for input and output
- 9.1.4. The standard streams

3. BOOKS RECOMMENDED FOR READING AND REFERENCE

Main Reading

Ref 1: Ivor Horton's Beginning Java™ 2, JDK™ 5 edition by Ivor Horton, Wiley India (P) Ltd India, 2006. ISBN 81-265-0570-2

Ref 2: Program Design, 4th edition by Peter Juliff, Prentice Hall India, 2002. ISBN-81-203-1622-3.

Recommended Links

- Official Sun Java site (Downloadable)
  http://java.sun.com/docs/books/tutorial/index.html
- Thinking in Java (3rd Edition) – Bruce Eckel (Downloadable)
  http://www.mindview.net/Books/TIJ/

4. PLATFORM

Any standard PC with a standard Java Compiler (JDK 1.5) running on a Windows/Linux platform. A visual development toolkit may be optionally used.

Note: *Under the detailed syllabus, page numbers of relevant text are given for each topic only as a guideline for minimal references based on the recommended main reading. These references are generally sufficient to understand the concepts and measure the expected depth of the content.*