

**IT2205: PROGRAMMING I****(Compulsory)****INTRODUCTION**

This is one of the 4 modules designed for Semester 2 of Bachelor of Information Technology Degree program.

**CREDITS: 04****LEARNING OUTCOMES**

On completion of this course, students will be able to design and develop programs for specified tasks using Java as an Object Oriented Programming Language.

**MINOR MODIFICATIONS**

When minor modifications are made to this syllabus, those will be reflected in the Virtual Learning Environment (VLE) and the latest version can be downloaded from the relevant course page of VLE. Please inform your suggestions and comments through the VLE. <http://vle.bit.lk>

**ONLINE LEARNING MATERIALS AND ACTIVITIES**

You can access all learning materials and this syllabus in the VLE: <http://vle.bit.lk>, if you are a registered student of BIT degree program. It is very important to participate in learning activities given in the VLE to learn this subject.

**ONLINE ASSIGNMENTS**

The assignments consist of two quizzes, assignment quiz 1 (It covers the first half of the syllabus) and assignment quiz 2 (It covers the second half of the syllabus). Maximum mark for a question is 10, minimum mark for a question is 0 (irrespective of negative scores). Final assignment mark is calculated considering 40% of assignment quiz 1 and 60% of assignment quiz 2. Pass mark for the online assignments in a course is 50. You are advised to do online assignments before the final exam of the course. It is compulsory to pass all online assignments to partially qualify to obtain year 1 certificate.

**FINAL EXAMINATION**

Final exam of the course will be held at the end of the semester. Each course in the semester 2 is evaluated using a two hour question paper which consists of 40-60 MCQs.

**OUTLINE OF SYLLABUS**

Topic	Minimum number of hours
• 1 - The history and evolution of Java	2

• 2 - Interacting with Java Programming environment	2
• 3 - Fundamentals of Java Programming	15
• 4 - Computer program design	5
• 5 - Object Orientation	13
• 6 - Packages and interfaces	5
• 7 - Exception handling	5
• 8 - Enumerations, Autoboxing and annotations	5
• 9 - Understanding generics	2
• 10 -Overview of some packages of the Java library	6
<b>Total for the subject</b>	<b>60</b>

## REQUIRED MATERIALS

### Main Reading:

**Ref 1:** Java: The Complete Reference <sup>TM</sup>, Eighth Edition by Herbert Schildt, Tata McGraw-Hill Edition 2011

**Ref 2:** Computer Program Design by Elizabeth A. Dickson, Tata McGraw-Hill Edition 2002

### Supplementary Reading:

- Java SE Development Kit 7 Downloads  
<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

## 2. DETAILED SYLLABUS

### 1. The history and evolution of Java (2hrs)

#### Instructional Objectives

- Explain the history of Java
- Describe the features of Java
- Briefly describe the evolution of Java

#### Sub Topics

- 1.1. Java's Lineage [Ref 1: Page 3]
- 1.2. The birth of Modern Programming : C and C++ [Ref 1: Page 4- 5]
- 1.3. The creation of Java and Microsoft's C# [Ref 1: Page 6 - 8]
- 1.4. Java and Internet [Ref 1: Page 8- 9]
  - 1.4.1. Java Applets

- 1.4.2. Security
- 1.4.3. Portability
- 1.5. The Bytecode [Ref 1: Page 9]
- 1.6. Java on the server Side [Ref 1: Page 10]
- 1.7. Features of Java [Ref 1: Page 10 - 13]
- 1.8. Evolution of Java [Ref 1: Page 13-14]
- 1.9. Understanding of the Java innovations [Ref 1: Page 16]

## **2. Interacting with Java Programming environment (2 hrs)**

### **Instructional Objectives**

- Install and setup the Java programming environment
- List the steps of writing a program
- Describe the compiling and interpretation process
- Compile and execute a Java program
- Describe how to use blocks of code and issues of lexical in Java
- Name keywords used in Java

### **Sub Topics**

- 2.1. Installing and setting the Java environment in one's computer teachers note
- 2.2. First few programs [Ref 1: Page 23 - 33]
  - 2.2.1. Writing and saving the Java source file
  - 2.2.2. How to compile and execute a Java program
  - 2.2.3. Structure of a Java program
  - 2.2.4. Using blocks of code and issues of lexical in Java at a glance
    - 2.2.4.1. Block of code
    - 2.2.4.2. Whitespace
    - 2.2.4.3. Identifiers
    - 2.2.4.4. Literals
    - 2.2.4.5. Comments
    - 2.2.4.6. Separators
  - 2.2.5. Java keywords [Ref 1: Page 33]

## **3. Fundamentals of Java Programming (15hrs)**

### **Instructional Objectives**

- Identify the appropriate data types and variables for representing different data
- Apply different operators to Java Programs
- List the operators according to their precedence levels
- Explain Java Programs by using different types of arrays
- Describe the use of different flow control statements
- Compare the differences of flow control statements
- Write Java programmes to solve problems using Flow Controls

### **Sub Topics**

- 3.1. Data types and variable in Java
  - 3.1.1. Data types [Ref 1: Page 35-43]

- 3.1.1.1. Java is a strongly typed language
    - 3.1.1.2. The primitive types
    - 3.1.1.3. A closer look at literals
      - 3.1.1.3.1. Integer literals
      - 3.1.1.3.2. Floating point literals
      - 3.1.1.3.3. Boolean literals
      - 3.1.1.3.4. Character literals
      - 3.1.1.3.5. String literals
  - 3.1.2. Variables [Ref 1: Page 44 - 48]
    - 3.1.2.1. Variable declarations
    - 3.1.2.2. Conversions in naming variables
    - 3.1.2.3. Dynamic initialization
    - 3.1.2.4. The scope and life time of variables
    - 3.1.2.5. Type conversion and casting
    - 3.1.2.6. Automatic type promotion in expressions
    - 3.1.2.7. The type promotion rules
- 3.2. Operators [Ref 1: Page 61 - 79]
  - 3.2.1. Arithmetic operators
  - 3.2.2. Increment and decrement
  - 3.2.3. Relational operators
  - 3.2.4. Boolean logical operators
  - 3.2.5. The bitwise operators
  - 3.2.6. The assignment operator
  - 3.2.7. The?: operator
  - 3.2.8. Operator precedence
  - 3.2.9. Usage of parenthesis
- 3.3. Control statements [Ref 1: Page 81 - 106]
  - 3.3.1. Selection statements
    - 3.3.1.1. If
    - 3.3.1.2. Switch
  - 3.3.2. Iteration statements
    - 3.3.2.1. For
    - 3.3.2.2. Do while
    - 3.3.2.3. While
    - 3.3.2.4. Nested loops
  - 3.3.3. Enhanced for loop
  - 3.3.4. Jump statements
    - 3.3.4.1. Break
    - 3.3.4.2. Continue
    - 3.3.4.3. Return
- 3.4. Arrays [Ref 1: Page 51 - 58]
  - 3.4.1. One dimensional
  - 3.4.2. Multi dimensional
  - 3.4.3. Using command line arguments [Ref 1: Page 154 - 159]

#### 4. Computer program design (5 hrs)

##### Instructional Objectives

- Describe what is an algorithm
- Explain the important steps of program design
- List different tools for program design

- Describe program structure and modular design

### **Sub Topics**

- 4.1. Overview of an algorithm
- 4.2. Program design
  - 4.2.1. Introduction to program design [Ref 2: Page 1 - 7]
  - 4.2.2. Flowcharts and design structures [Ref 2: Page 13 - 16]
  - 4.2.3. Pseudocode [Ref 2: Page 29 - 42]
  - 4.2.4. Other design tools [Ref 2: Page 345 - 356]
  - 4.2.5. Modules , flags and priming reads [Ref 2: Page 49 - 60]
  - 4.2.6. Structured design and Interactive programs [Ref 2: Page 105 - 116]
  - 4.2.7. Sorting [Ref 2: Page 267 - 269]
  - 4.2.8. Merging and matching two input files [Ref 2: Page 283 - 288]
  - 4.2.9. File updates [Ref 2: Page 303 - 317]
  - 4.2.10. Non sequential files [Ref 2: Page 323 - 328]

## **5. Object Orientation (13 hrs)**

### **Instructional Objectives**

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

### **Sub Topics**

- 5.1. Introducing classes [Ref 1: Page 109 - 126]
- 5.2. A closer look at methods and classes [Ref 1: Page 129 - 138]
- 5.3. Recursions [Ref 1: Page 139]
- 5.4. Understanding of object orientation features [Ref 1: Page 17 - 18]
- 5.5. Introducing access control [Ref 1: Page 141]
- 5.6. Understanding static and final key words [Ref 1: Page 145 - 146]
- 5.7. Nested and inner classes [Ref 1: Page 149 - 151]
- 5.8. Inheritance [Ref 1: Page 161 - 185]

## **6. Packages and interfaces (5 hrs)**

### **Instructional Objectives**

- Create packages in Java programs
- Implement and apply interfaces in Java programs

### **Sub Topics**

- 6.1. Packages [Ref 1: Page 187 - 194]
  - 6.1.1. Defining a package
  - 6.1.2. Finding packages and CLASSPATH
  - 6.1.3. Access protection
  - 6.1.4. Importing a package
- 6.2. Interfaces [Ref 1: Page 196 - 205]

- 6.2.1. Defining an interface
- 6.2.2. Implementing interfaces
- 6.2.3. Nested interfaces
- 6.2.4. Applying interfaces
- 6.2.5. Variables in interfaces

## **7. Exception handling (5 hrs)**

### **Instructional Objectives**

- Identify exception in Java program
- Handle errors in a program
- Write robust programs
- Process exceptions in Java programs

### **Sub Topics**

- 7.1. Exception handling fundamentals [Ref 1: Page 207]
- 7.2. Exception types [Ref 1: Page 208]
- 7.3. Uncaught exceptions [Ref 1: Page 209]
- 7.4. Using exceptions
  - 7.4.1. Try [Ref 1: Page 210 - 212]
    - 7.4.1.1. Nested try statements [Ref 1: Page 214]
  - 7.4.2. Catch [Ref 1: Page 210 - 212]
    - 7.4.2.1. Multiple catch clauses [Ref 1: Page 212]
  - 7.4.3. Throw [Ref 1: Page 216]
  - 7.4.4. Throws [Ref 1: Page 217]
  - 7.4.5. Finally [Ref 1: Page 218]
  - 7.4.6. Java's built in Exceptions [Ref 1: Page 220]
  - 7.4.7. Creating one's own exception sub classes [Ref 1: Page 221]
  - 7.4.8. Chained exceptions [Ref 1: Page 224]
  - 7.4.9. Newly added exception features of JDK new versions [Ref 1: Page 225]

## **8. Enumerations, autoboxing and annotations (5 hrs)**

### **Instructional Objectives**

- Describe what is enumerations
- Describe different type wrappers
- Explain what is autoboxing and how to use it.
- Discuss what are annotations and how to use it.

### **Sub Topics**

- 8.1. Enumerations
  - 8.1.1. Enumeration fundamentals [Ref 1: Page 259 - 261]
  - 8.1.2. The values() and valueOf() methods [Ref 1: Page 262 - 263]
- 8.2. Type wrappers [Ref 1: Page 268 - 269]
- 8.3. Autoboxing
  - 8.3.1. Autoboxing and methods [Ref 1: Page 271 - 272]
  - 8.3.2. Autoboxing/ unboxing occurs in expressions [Ref 1: Page 272 -273]

- 8.3.3. Boolean and character values [Ref 1: Page 274]
- 8.3.4. Autoboxing and preventing errors [Ref 1: Page 275]
- 8.4. Annotations
  - 8.4.1. Basics [Ref 1: Page 276]
  - 8.4.2. Specifying a retention policy [Ref 1: Page 276 -277]
  - 8.4.3. Obtaining annotations at run time by use of reflection [Ref 1: Page 277 - 281]
  - 8.4.4. Built in annotations[Ref 1: Page 286 - 287]

## **9. Understanding generics (2 hrs)**

### **Instructional Objectives**

- Explain generics in Java

### **Sub Topics**

- 9.1. What are generics [Ref 1: Page 326]
- 9.2. Generics work only with objects [Ref 1: Page 330]
- 9.3. Generic types differ based on their type arguments [Ref 1: Page 330]
- 9.4. How generics improve type safety [Ref 1: Page 330]
- 9.5. A general form of a generic class [Ref 1: Page 334]

## **10. Overview of some packages of the Java library (6 hrs)**

### **Instructional Objectives**

- Discuss different modules of String handling package
- Explain input and output used in Java input output package
- Explain applet packages and usage of its methods.

### **Sub Topics**

- 10.1. String handling [Ref 1: Page 371 - 395]
  - 10.1.1. The string constructors
  - 10.1.2. String length
  - 10.1.3. Special string operations
  - 10.1.4. Character extractions
  - 10.1.5. String comparisons
  - 10.1.6. Searching a string
  - 10.1.7. Modifying a string
  - 10.1.8. Data conversion using valueOf()
  - 10.1.9. Changing the case of a character within a string
  - 10.1.10. Additional string methods
  - 10.1.11. String buffer
  - 10.1.12. String builder
- 10.2. Introduction to collection framework and java.util package
  - 10.2.1. Using Scanner class [Ref 1: Page 563 - 564]
  - 10.2.2. Collection framework
    - 10.2.2.1. Collection overview [Ref 1: Page 453 – 454]

- 10.2.2.2. The collection interfaces [Ref 1: Page 456 - 463]
  - 10.2.2.3. The collection classes [Ref 1: Page 465 - 476]
  - 10.2.2.4. Accessing a collection via an iterator [Ref 1: Page 476 - 477]
- 10.3. Java input and output basics in java.io package
  - 10.3.1. Input and output basics [Ref 1: Page 289 - 290]
  - 10.3.2. Byte stream and character stream [Ref 1: Page 290]
  - 10.3.3. The predefined streams [Ref 1: Page 292]
  - 10.3.4. Reading console input [Ref 1: Page 293 - 295]
  - 10.3.5. Writing console output [Ref 1: Page 296]
  - 10.3.6. The printwriter class [Ref 1: Page 296]
  - 10.3.7. Reading and writing files [Ref 1: Page 297 - 302]
  - 10.3.8. Automatically closing a file [Ref 1: Page 303- 306]
- 10.4. java.applet class
  - 10.4.1. Two types of applets [Ref 1: Page 687]
  - 10.4.2. Applet basics [Ref 1: Page 687 - 688]
  - 10.4.3. The applet class [Ref 1: Page 688]
  - 10.4.4. Applet architecture [Ref 1: Page 690 - 691]
  - 10.4.5. Applet skeleton [Ref 1: Page 691]
  - 10.4.6. Applet initialization and termination [Ref 1: Page 692 - 693]
  - 10.4.7. Simple applet display methods [Ref 1: Page 693 - 695]
  - 10.4.8. The HTML applet tag [Ref 1: Page 699 - 701]

## PLATFORM

Any standard PC with a standard Java Compiler (JDK 1.7) 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.