**Riphah International Colleges**
A Project of Riphah International University
**Course Outline**

| | |
|---|---|
| **Course Code** | CS3623 |
| **Course Title** | **Advanced Computer Programming** |
| **Credit Hours** | 3(2+1) |
| **Prerequisites** | Object Oriented Programming |
| **Co-Requisites** | - |
| **Assessment Instruments with Weights** (homework, quizzes, midterms, final, programming assignments, lab work, etc.) | Quizzes + Assignments: 10<br>Project+ Presentation 10<br>Lab : 15<br>Mid Term 25<br>Final Exam: 40 |
| **Course Description** | In this course students will experience the Java Standard Edition J2SE of superior levels of product development using java. This course will help the students to create effective, scalable, maintainable, and editable applications to solve real life problems. |
| **Textbook** | No textbook, however following are good references<br>1. Head First Java, 2nd Edition, by Kathy Sierra, O' Really Publishers<br>2. Java How to Program, 9th Edition, by Deitel and Deitel<br>Effective Java (2nd Edition), by Joshua Bloch<br>3. Thinking in Java, 3rd Edition by Bruce Eckel. |
| **Reference Material** | Lecture Handouts, Sample Programs |
| **Course Goals/Objectives** | **Goal:** The primary goal of this course is to equip students with essential advanced java programming skills required for professional software development.<br>**Learning Objectives:** Students should be able to<br>a. Understand the use of IDEs to write and debug programs efficiently<br>b. Develop medium to advanced applications using previously learnt OO programming concepts<br>c. Build medium to advanced GUIs using swing<br>d. Understand java event model<br>e. Understand the concept of Threads and Thread Synchronization<br>f. Appreciate the usefulness of Wrapper, String and utility classes of Java API.<br>g. Use Java Collection API.<br>h. Understand and implement serialization of java objects.<br>i. Use features of the I/O API.<br>j. Perform database queries and updates using JDBC.<br>k. Understand and use Lambda Expressions<br>l. Write TCP/IP Client Server applications using sockets.<br>m. Execute methods on a remote object using RMI. |

| | |
|---|---|
| | n. Document and package a Java application. |

| Code # | Bloom's Taxonomy And Course Learning Outcomes | Course Teaching Strategies | Course Assessment Methods | %Weight |
|---|---|---|---|---|
| **1.0** | **Remember** | | | |
| 1.1 | Describe the concept of Java programming, how to utilize the java coding on real life problems. | Lectures | Quiz, Assignment | 20% |
| **2.0** | **Understand** | | | |
| 2.1 | Able to optimize real life problems using J2SE | Lectures, Case Studies | Quiz, Assignment | 10% |
| **3.0** | **Apply** | | | |
| 3.1 | Implement console and GUI based programs using J2SE | Lectures, Labs, Activities | Quiz, Assignment | 10% |
| 3.2 | Solve real life problems using J2SE with object programming | Lectures, Labs, Activities | Quiz, Assignment | 10% |
| 3.3 | Implement database connectivity with any DBMS | Lectures, Labs, Activities | Quiz, Assignment | 10% |
| 3.4 | Implement TCP/UDP connection in network programming using J2SE | Lectures, Labs, Activities | Quiz, Assignment | 10% |
| **4.0** | **Analyze** | | | |
| 4.1 | Analyze problem requirements to recognize what type of data and modules are involved in solution. | Lectures, Class Activities | Assignment, Project | 15% |
| **5.0** | **Create** | | | |
| 5.1 | Develop interactive GUI based application in J2SE of real life problems. | Lectures, Case Studies, Class Activities, Labs | Quiz, Assignment, Midterm, Project | 15% |

**Topics Covered in the Course, with Number of Lectures on Each Topic**

| Topic | Number of Lectures |
|---|---|
| Introduction to IDE, Introduction to java and its component | 2 (3 hrs) |
| Java programming fundamentals, and elementary programming | 1 (1.5 hrs) |
| Review of OO programming using java, Inheritance, Polymorphism, Encapsulation, and Abstract and Interface classes | 3 (4.5hrs) |
| Exception handling in java Checked, and Un-checked Exception, static and final packages | 2 (3 hrs) |
| I/O to Files and Console, Object Serialization | 3 (4.5 hrs) |
| Graphical User interface (GUI) and event handling | 2 (3 hrs) |
| JDBC | 4 (6 hrs) |
| Using Lambda Expressions | 2 (3 hrs) |
| Concurrency and Threading | 3 (4.5 hrs) |

| | |
|---|---|
| Network programming using sockets | 4 (6 hrs) |
| Distributed Applications using RMI | 4 (6 hrs) |
| Documenting and packaging java applications | 1 (1.5 hrs) |

**Weekly Lecture Plan**

| Week | Lecture | Topics | CLO# |
|---|---|---|---|
| **1** | **1** | Introduction To IDE, Introduction to java and its component | **1.1** |
| | **2** | Brief history of Java, Features/characteristics of Java, Java compilation Process, Types of Java Applications, Java Development Kit, Java Editions, and Java Development Tools | **1.1** |
| **2** | **3** | Difference between JRE, JDK, JVM, and JIT, Java Execution Flow | **1.1** |
| | **4** | Compiling and Execution of Java Program through Command Prompt, path and CLASSPATH, Anatomy of Java Program | **1.1** |
| **3** | **5** | To obtain input from the console using the Scanner class<br>To obtain input using the JOptionPane input dialog boxes<br>To use identifiers to name variables, constants, methods, and classes<br>To use constants to store permanent data<br>To declare Java primitive data types: byte, short, int, long, float, double, and char<br>To use Java operators to write numeric expressions | **1.1,** |
| | **6** | To use short hand operators<br>To cast value of one type to another type<br>To represent a string using the String type<br>To demonstrate defining classes and creating objects<br>To create objects using constructors<br>To access objects via object reference variables<br>To define a reference variable using a reference type<br>To access an object's data and methods using the object member access operator (.)<br>To define data fields of reference types and assign default values for an object's data fields | **1.1** |
| **4** | **7** | Review of OO programming using java, Inheritance, Polymorphism | **1.1, 2.1, 3.2** |
| | **8** | Encapsulation, and Abstract and Interface classes | **1.1, 2.1, 3.2** |
| **5** | **9** | Abstract method and an Abstract class, Interface, Interface as a Type, Interface vs. Class, defining an Interface, Implementing an Interface, Implementing multiple Interface's, Inheritance among Interface's | **1.1, 3.2** |
| | **10** | Overview of exceptions and exception handling, To explore the advantages of using exception handling, To write a try-catch block to handle exceptions, To use the finally clause in a try-catch block, Checked vs Unchecked | **1.1** |

| | | | |
|---|---|---|---|
| | | Exceptions, To declare exceptions in a method header, To throw exceptions in a method, To re-throw exceptions in a catch block, To define custom exception classes | |
| **6** | **11** | File and I/O Stream | **1.1, 3.2** |
| | **12** | Object Serialization and Externalization | **1.1, 2.1, 3.2** |
| **7** | **13** | Graphical User Interface (GUI) using Swing, GUI Helper classes, Creation of GUI Components | **1.1, 2.1, 3.1** |
| | **14** | Frames, Layout Manager, Layout Manager Types, Flow Layout, Grid Layout, Border Layout, Multiple Panels in a Frame, JPanel and Graphics, Multiple Panels in a Frame, Color Class, | **1.1,2.1,3.1** |
| **8** | **15** | Procedural vs Event-Driven Programming, Event classes, Event Information, Selected User Actions, | **1.1,3.2** |
| | **16** | Delegation Model, Inner class listener, The delegation model with inner class listener, Anonymous inner classes, Event Handler. Using coding | **1.1,3.2** |
| | | **MID TERM EXAM** | |
| **9** | **17** | Building GUI Through Drag And Drop, Use Of Event Listeners | **1.1,3.2** |
| | **18** | Building Fully Functional GUIs With Event Listeners And Layout Managers, Java Wrapper Classes, Writing Programs Using String And Associated Classes And Important String Functions | **1.1,3.2** |
| **10** | **19** | JDBC, Relational Databases, ODBC, SQL Statements | **3.3,1.1** |
| | **20** | Inserting, Updating | **3.3,1.1** |
| **11** | **21** | Deleting Records, Meta Data | **3.3,1.1** |
| | **22** | JDBC Revision | **3.3,1.1** |
| **12** | **23** | Using Lambda Expressions | **1.1** |
| | **24** | Example of Lambda Expression | **1.1** |
| **13** | **25** | Concurrency and Threading, single thread | **1.1,2.1** |
| | **26** | Multi-threading | **1.1,2.1** |
| **14** | **27** | Network Programming TCP | **3.4,1.1** |
| | **28** | Network Programming UDP | **3.4,1.1** |
| **15** | **29** | Distributed Applications using RMI | **1.1** |
| | **30** | Documenting and packaging java applications | **1.1** |
| **16** | **31** | Project Presentation | **1.1,2.1,3.1,3.2,4.1,5.1** |
| | **32** | Project Presentation | **1.1,2.1,3.1,3.2,4.1,5.1** |
| | | **FINAL TERM EXAM** | |