

# **Java Programming**

## **[Lab Manual]**

2015-2016



**INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN  
(IGDTU)**

<b>Version No.</b>	<b>Comments</b>	<b>Modification Date</b>	<b>Modification Done By</b>
1.0	Created Lab planner – added syllabus, installation guide	1 <sup>st</sup> Aug. 2011	Ankita Assistant Professor, Information Technology Dept., IGDTU.
2.0	Added List of Experiments and Bonus Problems	10 <sup>th</sup> Aug. 2011	Ankita Assistant Professor, Information Technology Dept., IGDTU.
3.0	Added MCA Syllabus, Linux installation guide and Advanced Problems	25 <sup>th</sup> July 2012	Ankita, Assistant Professor, Information Technology Dept., IGDTU
4.0	Added problems based on swing and jdbc	3 <sup>rd</sup> August 2015	Ankita, Assistant Professor, Information Technology Dept., IGDTU

## Table of Contents

<b>0.</b>	<b>Guidelines.....</b>	<b>2</b>
<b>1.</b>	<b>Syllabus.....</b>	<b>4</b>
<b>2.</b>	<b>List of Experiments</b>	
	<b>Core Java Basics</b>	
	2.1 Setting Up Your Machine for Java Development	
	2.2 Exercise based on classes, objects, arrays, command-line arguments, Strings.	
	2.3 Exercise based on Inheritance and Polymorphism	
	2.4 Exercise based on Interface, Packages and Exception Handling	
	2.5 Exercise based on Threading and File handling	
	<b>Advance concepts</b>	
	<b>RMI, JDBC, Applet, Event Handling, AWT Control &amp; Swing</b>	
	2.6 Problems based on Collection, Applets, Event handling & AWT Control	
	2.7 Problems based on RMI, Swing & JDBC	
<b>3.</b>	<b>Bonus Problems.....</b>	<b>22</b>

## **0. Guidelines**

This document titled “Java Lab Manual” is based on following guidelines:

1. Each Lab work has some objective to accomplish.
2. It is expected and strongly encouraged that students utilize their Lab hours efficiently.
3. Every effort should be made to complete the lab work in the assigned lab only.
4. d. Every student work on assigned system only. Any problem in the system must be reported to the lab in charge in advance. Reporting and ensuring the physical condition of the system is the responsibility of the students. No exercise in this regard will be entertained.
5. Any student found indulged in malpractices, disobedience will be debarred from the lab without any warning for a part/whole semester.

# 1. SYLLABUS

## **Paper: Java Programming**

### **PRE-REQUISITES:**

- Basic Object Oriented Programming Concepts

### **UNIT – I**

**Introduction to Java:** Characteristics of Java, Types of Java programs, JRE, about JVM, inner architecture of JVM, JDK

**Elements of Java Language:** Keywords, constants, variables and Data Types, Operators and Expressions, Decision Making, Branching and Looping: if..else, switch,?: operator, while, do, for statements, labeled loops, and jump statements: break, continue return.

**Introducing to classes, objects and methods:** Defining a class, adding variables and methods, Creating objects, constructors, class inheritance, Arrays and String: Creating an array, one and two dimensional arrays, string array and methods, Classes: String and String Buffer classes, Wrapper classes: Basics types, using super, Multilevel hierarchy abstract and final classes, Object class, Packages and interfaces, Access protection, Extending Interfaces, packages.

### **UNIT – II**

**Exception Handling:** Fundamentals exception types, uncaught exceptions, throw, throw, final, built in exception, creating your own exceptions,

**Multithreaded Programming:** Fundamentals, Java thread model: priorities, synchronization, messaging, thread classes, Runnable interface, inter thread Communication, suspending, resuming and stopping threads.

**Input/Output Programming:** Basics, Streams, Byte and Character Stream, predefined streams, Reading and writing from console and files.

**Using Standard Java Packages (lang, util, io, net).** Networking: Basics, networking classes and interfaces, using java.net package, doing TCP/IP and Datagram Programming,

**The Collection Framework:** The Collection Interface, Collection Classes, Working with Maps & Sets

### **UNIT – III**

**Event Handling:** Different Mechanism, the Delegation Event Model, Event Classes, Event Listener Interfaces, Adapter and Inner Classes, Working with windows, Graphics and Text, using AWT controls, Layout managers and menus, handling Image, animation, sound and video, Java Applet.

**Swing:** Introduction to JFC (Java Foundation Classes), Features of Swing, Comparison with AWT, Advanced Control.

### **UNIT-IV**

**Accessing database using JDBC:** Introduction to DBMS & RDBMS, JDBC API, JDBC Application Architecture, Obtaining a Connection, JDBC Models: Two Tier and Three Tier Model, ResultSet, Prepared Statement, Callable Statement.

**RMI (Remote Method Invocation):** Introduction, Steps in creating a Remote Object, Generating Stub & Skeleton, RMI Architecture, RMI packages.

**Java Bean:** Introduction, Bean Architecture, Using the Bean Development Kit, Creating simple bean-properties, methods and events, Packing beans- the manifest & the jar, Java bean package, Introduction to NetBean-Overview.

## 2.1 LAB ASSIGNMENT

### Java Basics

**Objective:** At the end of the assignments, students will be able to:-

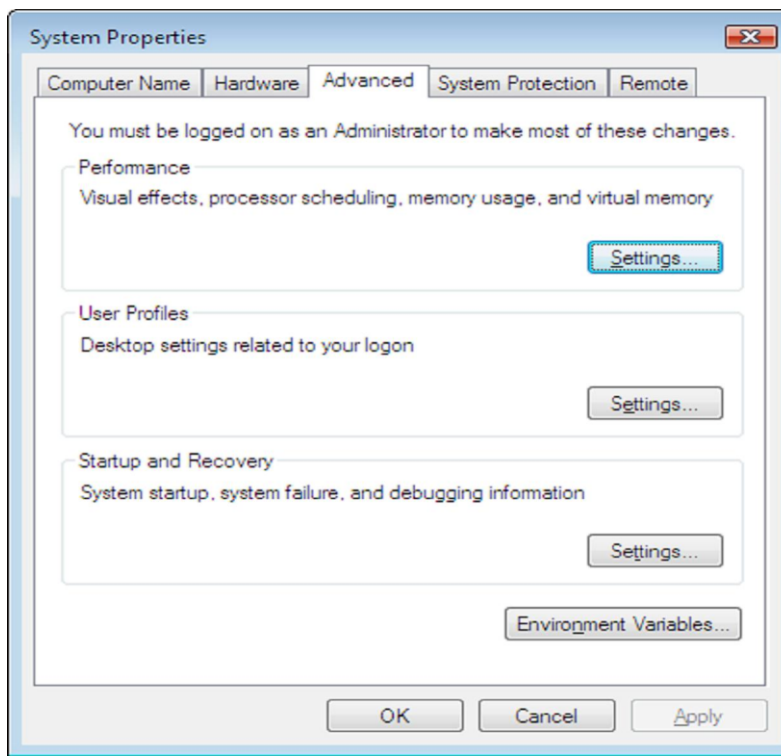
- **Setting Up Your Machine for Java Development**
- **Compile and run a simple program in java.**
- **Create classes and write programs using objects, arrays, command-line arguments, Strings.**

#### Setting Up Your Machine for Java Development

##### ➤ **Windows Platform:**

To be able to write, compile and execute Java applications, you need a set of tools called Java Software Development Kit (JSDK). This SDK could be obtained from Java official site at <http://java.sun.com>.

After downloading the installation file (about 60 MB) you are ready to start the installation process. Follow the procedure to prepare your machine. After Installation of JSDK software you need now to define some environment variables to be able to use the tools you've just installed. To define an environment variable, right-click on *My Computer* icon on the desktop or start menu, then select *Properties* from the pop-up menu. The *System Properties* dialog appears, from the top of the dialog box select *Advanced* tab as shown. From *Advanced* tab, click on *Environment Variables* button, Environment Variables dialog appears as shown :

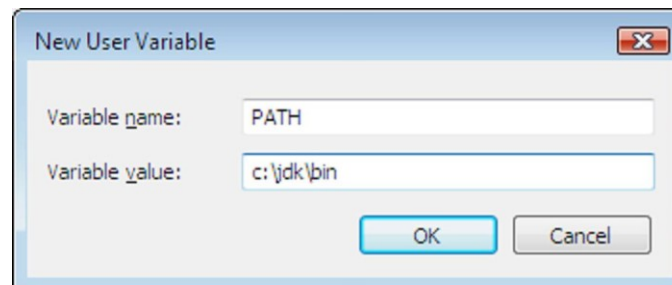
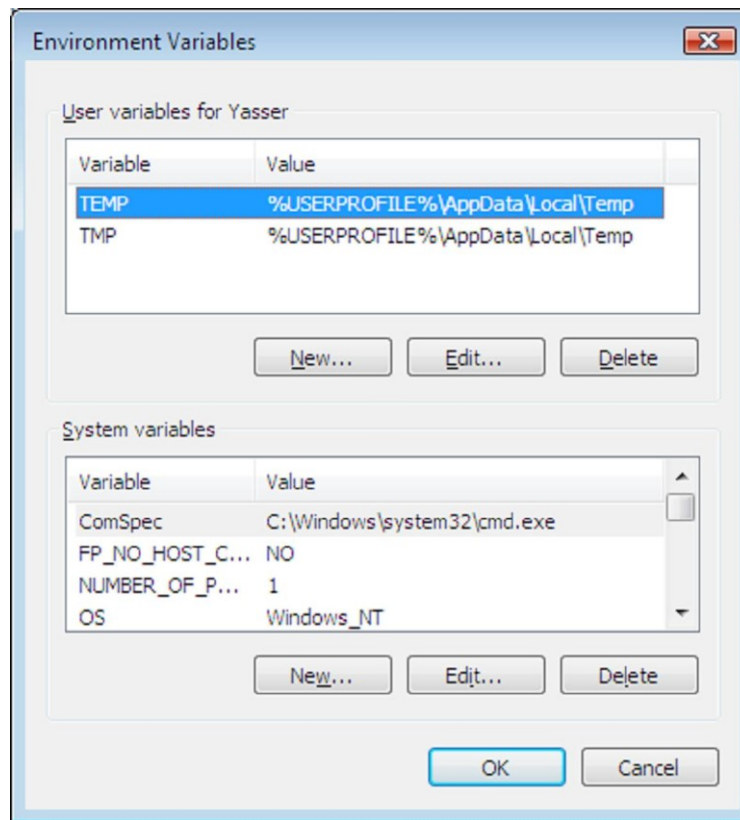


To add new environment variable, click on *New* button in *User Variables* field, note that you may not modify system variables unless you have administrator privileges. Once you click on *New* button, variable addition dialog appears as in figure 2.6.

In *Variable name* field enter the name of your variable, in *Variable value* field enter the value of the variable then click on *OK*. You need to add two variables to use with JDSK:

- **PATH**: points to bin directory in JSDK home directory (e. g. c:\jdk\bin)
- **CLASSPATH**: points to the folder in which you will create you Java programs (e. g. c:\Documents and Settings\Student\Desktop\java).

Your computer is ready now to be used for Java development.



## ➤ **Linux Platform**

### **Setup the environment variable**

You also need to setup JAVA\_HOME and PATH variable. Open your \$HOME/.bash\_profile or /etc/profile (system wide) configuration. Open your .bash\_profile file:

```
$ vi $HOME/.bash_profile
Append following line:
export JAVA_HOME=/usr/lib/jvm/java-6-sun
export PATH=$PATH:$JAVA_HOME/bin
Save and close the file.
```

### **Test your new JDK**

Type the following command to display version:

```
$ java -version
Output:
java version "1.6.0_03"
Java(TM) SE Runtime Environment (build 1.6.0_03-b05)
Java HotSpot(TM) Server VM (build 1.6.0_03-b05, mixed mode)
```

### **Try HelloWorld.java - first java program**

```
$ vi HelloWorld.java
```

## **Compile and Run a Java Program**

### ➤ **Windows Environment**

#### **Notepad:**

- Write a simple Java Program to display “HELLO WORLD” into Notepad. Save as the Java file with the java extension.
- Compile the <javafile> with javac command on DOS prompt.
- Execute the <classfile> with the java command on DOS prompt.

### ➤ **Linux Environment**

#### **VIM/VI Editor**

- Write a simple Java Program to display “HELLO WORLD”. Save as the Java file with the java extension.
- Compile the <javafile> with javac command on terminal.  
igit@igit-desktop:~\$ javac Firstprogram.java
- Execute the <classfile> with the java command on terminal.  
igit@igit-desktop:~\$ java Firstprogram



## 2.2 LAB ASSIGNMENT

### Classes & Objects

**Objective:** At the end of the assignments, students will be able to:-  
Create classes and write programs using objects, arrays, command-line arguments, Strings.

#### Java - Class, Object and Methods using Conditional statements and Loops

1. Write a program to display the prime numbers up to n.
2. Write a program to list all even numbers less than or equal to the number n. Take the value of n as input from user.
3. Define a class Rectangle with its length and breadth. Provide appropriate constructor(s), which gives facility of constructing rectangle object with default values of length and breadth as 0 or passing value of length and breadth externally to constructor. Provide appropriate accessor & mutator methods to Rectangle class. Provide methods to calculate area & to display all information of Rectangle.  
Design different class TestRectangle class in separate source file, which will contain main function. From this main function, create 5 Rectangle objects by taking all necessary information from the user.
4. Create a class Book which describes its Book\_title and Book\_price. Use getter and setter methods to get & set the Books description. Implement createBooks and showBooks methods to create n objects of Book in an array. Display the books along with its description as follows:-

Book Title	Price
Java Programming	Rs.350.50
Let Us C	Rs.200.00

Note: createBooks & showBooks should not be member functions of Book class.

5. Modify the program which is created in assignment 2 as follows  
The class has attributes **length** and **width**, each of which defaults to 1. It should have member functions that calculate the perimeter and area of the rectangle. It should have set and get functions for both length and width. The set functions should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0.
6. Create a class Date for manipulating dates. Provide a constructor that enables an object of this class to be initialized when it is declared (You can select any default values for the day, month & year, e.g. your birth date). Provide the necessary functionality to perform error checking on the initializer values for data members day,

month, and year. Also, provide a member function to add an integer in a date to obtain a new date.

Design separate class Employee which will have following information.

Employee Number	Number
Employee Name	Text
Joining Date	Date

Provide appropriate constructor(s) & methods to this class. Provide main function which will create 5 objects of Employee class.

7. Write a program that takes a String through Command Line argument and display the length of the string. Also display the string into uppercase and check whether it is a palindrome or not. (Refer Java API Documentation)

8. Write a program, which will take three strings from the command line and display the number of characters in each string.

9. Write a program that accepts two numbers from the Command Line and prints them out. Then use a **for loop** to print the next 17 numbers in the sequence where each number is the sum of the previous two. For example:

```
input> java prob2 1 3
output> 1 3 4 7 11 18 29 47 76 123 322 521 843 1364
```

10. Write a program that accepts two numbers in the range from 1 to 40 from the Command Line. It then compares these numbers against a single dimension **array** of five integer elements ranging in value from 1 to 40. The program displays the message **FOUND** if the two inputted values are found in the array element. For example:

```
input>java prob 3 29
output>Your first number was 3
Your second number was 29
Its Bingo! // this message if 3 and 29 is found in the array
Not Found! // this message if 3 and 29 is not found in the    //array
The array was 7 25 5 19 30
```

11. Write a program that allows you to create an integer **array** of 18 elements with the following values: **int A[]={3,2,4,5,6,4,5,7,3,2,3,4,7,1,2,0,0,0}**. The program computes the sum of element 0 to 14 and stores it at element 15, computes the average and stores it at element 16 and identifies the smallest value from the array and stores it at element 17.

12. Create a class **Term**. This class represents a term of a polynomial such as  $2x^4$  where 2 is coefficient and 4 is exponent of the term.

---

**Data members:-**

**int coefficient**

**int exponent**

Create another class **Polynomial**. The internal representation of a polynomial is an array of Terms. The size of this array should be fixed. Provide a constructor for this class that will set all terms of a polynomial object as zero (where coefficient is 0 and exponent is 0). Provide following functions:

setTerm(int, int) – Setting a term of a polynomial object. Each successive call of this function should set next term of the polynomial object.

It should do the following validations:-

Whether the exponent of the term being set is already used.

Whether the array size limit is exceeded.

Whether the exponent is negative.

In all the cases it should not set the term and display an appropriate message.

**sort()** – to arrange the terms in ascending order of exponents.

Provide a function to print a polynomial object

**13.** Write a program that accepts two numbers and a operator like (+, -, \*, /) as command line arguments and perform the appropriate operation indicated by operator. If the user enters any other character the appropriate message will be displayed. The output of the program should be displayed to the user.

## 2.3 LAB ASSIGNEMENT

### Inheritance and Polymorphism

**Objective:** At the end of the assignments, participants will be able to create abstract classes, create a new class by extending an existing class, write code to exhibit polymorphic behavior of a method call, use of interfaces

#### Concept: Inheritance and Polymorphism

1. Create an abstract class Instrument which is having the abstract function play. Create three more sub classes from Instrument which is Piano, Flute, Guitar. Override the play method inside all three classes printing a message

“Piano is playing tan tan tan tan ” for Piano class

“Flute is playing toot toot toot toot” for Flute class

“Guitar is playing tin tin tin ” for Guitar class

You must not allow the user to declare an object of Instrument class.

Create an array of 10 Instruments.

Assign different type of instrument to Instrument reference.

Check for the polymorphic behavior of play method.

Use the instanceof operator to print that which object stored at which index of instrument array.

2. Create an abstract class Compartment to represent a rail coach. Provide a abstract function notice in this class. Derive FirstClass, Ladies, General, Luggage classes from the compartment class. Override the notice function in each of them to print notice suitable to the type of the compartment.

Create a class TestCompartment . Write main function to do the following:

Declare an array of Compartment pointers of size 10.

Create a compartment of a type as decided by a randomly generated integer in the range 1 to 4. Check the polymorphic behavior of the notice method.

3. Create a class Medicine to represent a drug manufactured by a pharmaceutical company. Provide a function displayLabel() in this class to print Name and address of the company. Derive Tablet, Syrup and Ointment classes from the Medicine class. Override the displayLabel() function in each of these classes to print additional information suitable to the type of medicine. For example, in case of tablets, it could be “store in a cool dry place”, in case of ointments it could be “for external use only” etc.

Create a class TestMedicine . Write main function to do the following:

Declare an array of Medicine references of size 10. Create a medicine object of the type as decided by a randomly generated integer in the range 1 to 3.

Check the polymorphic behavior of the displayLabel() method.

4. Create a class Car which contains members speed, noOfGear. The class has a method drive() which is responsible to provide starting speed and noOfGears to a Car. Implement display() method which will display all attributes of Car class.

The class SportCar is derived from the class Car which adds new features AirBallonType. When this method is invoked, initial speed and gear status must be displayed on console. Override the display method which display all attribute of the SportCar. Make use of super class display() method.

## Interface, Packages and Exception Handling

## Concept: Interface & packages

**Attributes:**

### Member functions:

```
//person to be greeted by this greeter.
```

```
//person initialized earlier.
```

**Attributes:**

### Member functions:

```
//strings with atleast five advice messages
```

```
//list of messages and returns it to the caller of //this method
```

Page 14

For  
e.g.,  
java GreeterTest Mars Venus

then the program should  
    print Hello,  
    Mars!  
    Hello,  
    Venus!  
    Advice: Never say No

    Goodbye Venus!  
    Goodbye Mars!

2. Create a package `igit.roll.shape` containing the following classes and interfaces. An interface `Polygon` containing the members as given below:

    area           float  
    perimeter     float

void calcArea( );     abstract method to calculate area of a particular polygon given its dimensions

void calcPeri( );     abstract method to calculate perimeter of a particular polygon given its dimensions

void display( );     method to display the area and perimeter of the given polygon

Create a class `Square` that implements `Polygon` and has the following member:

    side     float  
    Square(float s);     constructor to initialize side of square

Create another class `Rectangle` that implements `Polygon` and has the following member:

    length     float  
    breadth    float  
    Rectangle(int len, int bre);     constructor to initialize length and breadth of a rectangle

Outside the package, create a class that imports the above package and instantiates an object of the `Square` class and an object of the `Rectangle` class. Call the above methods on each of the classes to calculate the area and perimeter given the side and the length/breadth of the `Square` class and the `Rectangle` class respectively.

## Concept : Exception Handling

### Basic Problems:

3. Create a class called CalcAverage that has the following method:

```
public double avgFirstN(int N)
```

This method receives an integer as a parameter and calculates the average of first N natural numbers. If N is not a natural number, throw an exception `IllegalArgumentException` with an appropriate message.

4. Create a class Number having the following features:

#### Attributes

int	first number(a)	
int	second number(b)	
double	result	//stores the result of arithmetic operations performed on a and b

#### Member functions

Number(x, y)	constructor to initialize the values of a and b
add()	stores the sum of a and b in result
sub()	stores difference of a and b in result
mul()	stores product in result
div()	stores a divided by b in result

Test to see if b is 0 and throw an appropriate exception since division by zero is undefined.

Display a menu to the user to perform the above four arithmetic operations.

5. Create a class `BankAccount` having the members as given below:

accNo	integer
custName	string
accType	string (indicates 'Savings' or 'Current')
balance	float

Include the following methods in the `BankAccount` class:

```
void deposit(float amt);  
void withdraw(float amt);  
float getBalance();
```

`deposit(float amt)` method allows you to credit an amount into the current balance. If amount is negative, throw an exception `NegativeAmount` to block the operation from being performed.



withdraw(float amt) method allows you to debit an amount from the current balance. Please ensure a minimum balance of Rs. 1000/- in the account for savings account and Rs. 5000/- for current account, else throw an exception InsufficientFunds and block the withdrawal operation. Also throw an exception NegativeAmount to block the operation from being performed if the amt parameter passed to this function is negative.

getBalance() method returns the current balance. If the current balance is below the minimum required balance, then throw an exception LowBalanceException accordingly.

Have constructor to which you will pass, accno, cust\_name, acctype and initial balance. And check whether the balance is less than 1000 or not in case of savings account and less than 5000 in case of a current account. If so, then raise a LowBalanceException. In either case if the balance is negative then raise the NegativeAmount exception accordingly.

**Advance Problems:**

- Write a program to implement a Queue, using user defined Exception Handling (also make use of throw, throws).
- WAP for sorting given list of names in ascending order

## 2.5 LAB ASSIGNMENT 4

### Threading and File handling

**Objective:** At the end of the assignments, participants will be able to create handle multithreading and file operations.

1. Write a Java program that demonstrates a high-priority thread using sleep to give Lower priority threads a chance to run.
2. Implement three classes: Storage, Counter and Printer. The Storage class should store an integer. The Counter class should create a thread and starts counting from 0 (0,1,2,3...) and stores each value in the Storage class. The Printer Class should create a thread that keeps reading the value in the Storage class and printing it. Write a program that creates an instance of the Storage class and set up a Counter and Printer object to operate on it. Identify that, whether synchronization is required or not in this assignment. If yes, implement it.
3. Write a Java application that will accept two filenames (text files) as command line arguments and use two threads to read contents from the two text files. Each of the threads should sleep for a random time after displaying filename with each line.
4. Write a java application that will create and start two threads. One thread will read a text file (Number.txt) containing five positive integers one on each line. The second thread should calculate factorial of the number read by the first thread and print the message on the screen as "Factorial of x is y", here x is number & y is factorial of the number. The two threads should work in synchronization. Handle all necessary exceptions.
5. Write a program which take source file and destination file as input as command line arguments. It copies the source file contents to destination file. If source file does not exist, it should give appropriate message to use. If destination file does not exist, it should be created. If it exists, program should ask that, "whether you want to overwrite? (Yes/No)". On the basis of user choice, appropriate action should be taken.

## 2.6 LAB ASSIGNMENT 5

### COLLECTION FRAMEWORK, APPLET, EVENT HANDLING & AWT CONTROL

**Objective:** At the end of the assignments, participants will be able to understand applet,event handling mechanism and AWT controls.

1. Write a program to illustrate the use of following collection framework classes & interface:  
List- ArrayList and LinkedList  
Set- HashSet and TreeSet  
Map-HashMap and TreeMap
2. Write a program to display a moving Banner in a Applet Window.
3. Write a program to handle Key and Mouse Events.
4. Write a simple program to display line, circle, rectangle, eclipse, arcs and polygon.
5. Write a program to display:

#### Indira Gandhi Institute of Technology Student Detail Form

<b>Name:</b>	<input type="text"/>
<b>Enrollment No:</b>	<input type="text"/>
<b>Roll No:</b>	<input type="text"/>
<b>Course:</b>	<input checked="" type="checkbox"/> <b>B.Tech</b> <input type="checkbox"/> <b>MCA</b>
<b>Semester:</b>	<input type="text"/>
<b>Address:</b>	<input type="text"/> <input type="text"/>
<b>Contact No:</b>	<input type="text"/>
	<input type="button" value="SUBMIT"/> <input type="button" value="RESET"/>

#### Advance Problems

- WAP that lets users create Pie charts. Design your own user interface (with Swings & AWT).
- WAP that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the + - x / % operations. Add a text field to display the result.

## 2.7 LAB ASSIGNMENT 6

### RMI, SWING & JDBC

**Objective:** At the end of lab session participants will be able to understand the advanced java concepts(RMI, Swing & JDBC)

1. WAP a calculator(+,-,\*,/) problem by using RMI(Remote method invocation).
2. WAP to display six buttons and image on a panel using JFrame.
3. WAP to display an image and a string in a label on the JApplet.
4. WAP to display calculator using swing
5. Write a java program to create a connection with emp database and then display the first 5 records from the table Employee. (Use Type-1 driver for MS-Access)
6. Write a java programs as follows:

Accept the empno from the user. Write a method display( ) which takes the empno as parameter and displays the detail information of the employee. If employee with the empno does not exist display appropriate message to user. WAP to insert data into the database using the prepared statement.

### 3.0 BONUS PROBLEMS

1. Create a class **Matrix**. Internal representation of this class will be a two dimensional array of size 10 by 10. In addition, the class should have following data members and member functions:

**Data members:-**

int **rows**

int **columns**

**Constructors**

**The default constructor**

Matrix() - This should set each of the array element to zero.

**Overloaded constructor**

Matrix(int, int) - This constructor should call the default constructor first. It should then assign the value of first parameter to variable **rows**, and the value of the second parameter to variable **columns**. You can assume that the values of both the parameters will be less than or equal to 10.

**Member functions**

void setElement(int r, int c, int value) - This function should set the array element at row r and column c to the value val. This assignment should be done only if val is positive, r and c are valid else the element should be set to zero.

Matrix transpose () – This function should transpose the matrix. Transpose of a matrix is another matrix where the elements in rows of the first matrix become elements of the corresponding columns in the new matrix.

Provide a function to print a Matrix object.

2. Create a class called **complex** for performing arithmetic operations with complex numbers. Use floating point variables to represent the private data of the class. Provide a default constructor that initializes the object with some default values. Provide public **member functions** for each of the following

Addition of two complex numbers: It returns the result obtained by adding the respective real parts and the imaginary parts of the two complex numbers.

Subtraction of two complex numbers: It returns the result obtained by subtracting the respective real parts and the imaginary parts of the two complex numbers.

display() – It displays the complex number in **a+bi** format. The output should be displayed as follows:-

Sum of  $a_1+b_1i$  &  $a_2+b_2i$  is :  $a_3+b_3i$

3. Write a stream based program which will accept Roll Number, Name, Age and Address from user .

- Age and Roll-no should be numeric. Handle with built-in exception.
- None of the field should be blank. Handle with custom exception, Ask user whether to write the data in the file. If answer is yes then data is saved into

a file as an object (User can write many records in the file), otherwise terminate the current program

4. Write another program to display all the records saved into the file
5. WAP to generate a set of random numbers. Find its sum and average. The program should also display '\*' based on the random numbers generated.
6. WAP to display label on a frame with the help of JFrame
7. WAP. To display an image and a string in a label on the JFrame.
8. Create a HTML page with CSS styling.
9. WAP to include HTML page into servlet which will have login page with following details

LOGIN

User id:	<input type="text"/>
Password:	<input type="password"/>
<input type="button" value="SUBMIT"/>	<input type="button" value="RESET"/>

10. WAP to include HTML page into JSP.
11. WAP that implements a simple client/server application. The client sends data to a server. The server receives the data, uses it to produce a result and then sends the result back to the client. The client displays the result on the console. For ex: The data sent from the client is the radius of a circle and the result produced by the server is the area of the circle.
12. WAP that reads a line of integers and then displays each integer and the sum of all integers. (use StringTokenizer class)

**References**

1. <http://www.scribd.com/doc/13161906/Java-Lab-Manual-With-Java-Installation-Guide>
2. <http://www.scribd.com/afiz2u/d/47440325-Complete-Java-PPT>
3. [http://books.google.co.in/books?id=qBKO1b8CsNUC&printsec=frontcover&source=gbbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](http://books.google.co.in/books?id=qBKO1b8CsNUC&printsec=frontcover&source=gbbs_ge_summary_r&cad=0#v=onepage&q&f=false)
4. Patrick Naughton and Herbertz Schildt, “Java-2: The Complete Reference”, TMH
5. Bruce Eckel, “Thinking in Java 4th Edition”
6. Other Web Resources