ALAGAPPA UNIVERSITY

[Accredited with 'A+' Grade by NAAC (CGPA:3.64) in the Third Cycle and Graded as Category–I University by MHRD-UGC]

(A State University Established by the Government of Tamil Nadu)

KARAIKUDI – 630 003

Directorate of Distance Education

M.Sc. [Computer Science]

I - Semester

341 14

LAB - ADVANCED JAVA PROGRAMMING

Reviewer	
Dr. K. Kuppusamy	Professor and Head (i/c), Department of Computational Logistics, Alagappa University, Karaikudi

Author

Dr. Kavita Saini, *Assistant Professor, School of Computer Science & Engineering, Galgotias University, Greater Noida.*

Vikas® is the registered trademark of Vikas® Publishing House Pvt. Ltd.

VIKAS® PUBLISHING HOUSE PVT. LTD.
E-28, Sector-8, Noida - 201301 (UP)
Phone: 0120-4078900 • Fax: 0120-4078999
Regd. Office: A-27, 2nd Floor, Mohan Co-operative Industrial Estate, New Delhi 1100 44
• Website: www.vikaspublishing.com • Email: helpline@vikaspublishing.com

Work Order No. AU/DDE/DE1-616/Printing of Course Materials/2019, Dated 19.11.2019 Copies - 200

LAB - ADVANCED JAVA PROGRAMMING

Syllabi

BLOCK 1: INTRODUCTION

1. Simple Java Program
2. Program using JDBC with Create, Insert Table Data
3. SQL Exception, SQL Warning

BLOCK 2 : INETADDRESS

4. Programs using TCP/IP Client Sockets, TCP/IP Server Sockets
5. Program with URL, URL Connection, Data Grams Connection
6. Client/Server Applications using RMI

BLOCK 3 : BDK AND DESIGN PATTERNS

7. Simple Programs using Bean Development Kit, JAR Files
8. Program with Design Patterns
9. Program with Events and Methods

BLOCK 4 : SERVLETS

10. Create a Servlet to Read the Parameters
11. Programs using Cookies
12. Programs with Session Tracking

BLOCK 5 : JAPPLET AND AWT CLASSES

13. Programs using JApplet, Buttons, Combo, Trees, Tables, Panes
14. Programs with AWT Classes, Working with Graphics, Color and Font

INTRODUCTION

NOTES

Java is a third generation programming language which implements the concept of Object-Oriented Programming (OOPs). It inherits many features of the existing languages, C and C++, along with the addition of new features, making it a simple object-oriented language that is also easy to learn. Java can either have single or compound statements. Java has control statements that are broadly classified into three categories, namely conditional statements, iteration statements and jump statements. The main objective of object-oriented programming is to present various real-world objects as program elements. All concepts related to object-oriented programming, such as data abstraction, encapsulation, inheritance and polymorphism, are implemented with the help of classes. Working with actual data requires a mechanism that deals with a collection of data items. In Java, different data types like arrays and vectors are offered to handle such collections.

This lab manual, *Advanced Java Programming*, contains several programs based on Java concepts, such as JDBC, TCP/IP client and server sockets, RMI, JApplet and AWT classes to provide the concept of programming. In addition, it will help students in coding and debugging their programs. The manual provides all logical, mathematical and conceptual programs that can help to write programs very easily in Java language. These exercises shall be taken as the base reference during lab activities for students. There are also many Try Yourself Questions provided to students for implementation in the lab.

LAB REQUIREMENTS

NOTES

To write and run a Java program, you need to install a software like J2SDK 1.7. SDK stands for system development kit. SDK is also known as JDK (Java Development Kit) which contains JRE (Java Runtime Environment). It provides a platform that enables the program to run on your computer.

Following are the steps given below that explains how to write and execute a Java program.

Step 1: Write a Java code using text editor (notepad).

1. Write a program to print Hello Java.

```
//main class
    public class Sample1
{
    public static void main(String args[])
{
System.out.println("Hello Java");
    }
}
```

Step 2: Save the file as **Sample1.Java**. We have named the file as Sample1, the thing is that we should always name the file same as the public classname. In our program, the public class name is Sample1. So, our file name should be **Sample1.Java**.

Step 3: Set environment variable.

Follow the steps to set the environment variable:

Right Click on MyComputer → Properties → Advanced System settings → Inside Advanced tab

Click Environment variables → Inside System Vaiables click New → Give variable name (For example var) → Give variable value. It is path in your system where Java compiler is available (For example variable value :C:\Program Files\Java\jdk1.6.0_23\bin). Inside bin Javac is Java compiler.

Click Ok.

Step 4: Go to command prompt by using Start → Run → cmd OR start → type cmd in search program and file.

Step 5: Write following command for compilation of program.

```
Javac Sample1.Java
```

Step 6: To run program, use the following command.

```
Java Sample1
```

NOTES

Output:

```
Hello Java
```

2. Write a program to add two integers and print it on the screen.

```
public class AddTwoIntegers
{

   public static void main(String[] args)
 {

        int first = 10;
        int second = 20;

        int sum = first + second;

System.out.println("The sum is: " + sum);
    }
}
```

Output:

```
The sum is: 30
```

3. Write a program to multiply two floating point numbers.

```
public class MultiplyTwoNumbers
 {

   public static void main(String[] args)
 {

        float first = 1.5f;
        float second = 2.0f;
```

```
        float product = first * second;

System.out.println ("The product is: " + product);
    }
}
```

NOTES

Output:

4. Write a program to swap two numbers using a temporary variable.

```
public class SwapNumbers
 {

   public static void main(String[] args)
{

        float first = 1.20f, second = 2.45f;

System.out.println("—Before swap—");
System.out.println("First number = " + first);
System.out.println("Second number = " + second);

        // Value of first is assigned to temporary
        float temporary = first;

        // Value of second is assigned to first
        first = second;
      // Value of temporary (which contains the initial
value of first) is assigned to second
        second = temporary;
System.out.println("—After swap—");
System.out.println("First number = " + first);
System.out.println("Second number = " + second);
    }
}
```

NOTES

Output:

```
--Before swap--
First number = 1.2
Second number = 2.45
--After swap--
First number = 2.45
Second number = 1.2
```

5. Write a program to print the largest number among the three numbers.

```
public class Largest
 {

   public static void main(String[] args)
{

        double n1 = -4.5, n2 = 3.9, n3 = 2.5;

if ( n1 >= n2 && n1 >= n3)
System.out.println(n1 + " is the largest number.");

        else if (n2 >= n1 && n2 >= n3)
System.out.println (n2 + " is the largest number.");

        else
System.out.println (n3 + " is the largest number.");
    }
}
```

Output:

```
3.9 is the largest number.
```

Try Yourself

1. Write a Java program to divide two numbers and print on the screen.
2. Write a Java program to print the result of the following operations.
   a. -5 + 8 * 6
   b. (55+9) % 9
3. Write a Java program to print the sum (addition), multiply, subtract, divide and remainder of two numbers.
4. Write a Java program to print the area and perimeter of a circle.

6. Write a program to demonstrate the implementation of class, object and constructor.

```
//main class
public class Puppy
{
// This constructor has one parameter, name.
   public Puppy(String name)
{
System.out.println("Passed Name is :" + name );
    }

   public static void main(String []args)
 {
 // Following statement would create an object myPuppy
      Puppy myPuppy = new Puppy ("tommy");
    }
}
```

Output:

```
Passed Name is :tommy
```

NOTES

7. Write a program to demonstrate how to access instance variables and methods of a class.

```
public class Puppy
 {
   int puppyAge;
  public Puppy(String name)
{
     // This constructor has one parameter, name.
System.out.println("Name chosen is :" + name );
 }
  public void setAge( int age )
{
puppyAge = age;
   }
  public int getAge( )
{
System.out.println("Puppy's age is :" + puppyAge );
     return puppyAge;
   }
  public static void main(String []args)
 {
     /* Object creation */
     Puppy myPuppy = new Puppy( "tommy" );
     /* Call class method to set puppy's age */
myPuppy.setAge( 2 );
    /* Call another class method to get puppy's age */
myPuppy.getAge( );
     /* You can access instance variable as follows as
well */
System.out.println("Variable Value :" + myPuppy.puppyAge
);
   }
}
```

NOTES

Output:

```
Name chosen is :tommy
Puppy's age is :2
Variable Value :2
```

8. Write a Java program to print first 10 numbers in Fibonacci series.

```
classFibo
{
public static void main(String args[])
    {
       inta,b,temp,n;
       a=0;
       b=1;
       for (n=1;n<=10;n++)
       {
       System.out.println(a);
       temp=a+b;
           a=b;
           b=temp;
       }
       }
         }
```

Output:

```
C:\javaexprg>javac Fibo.java

C:\javaexprg>java Fibo
0
1
1
2
3
5
8
13
21
34
```

NOTES

9. Write a Java program to print factorial of a given number.

```
    ImportJava.util.*;
    class Factorial
          {
    public static void main(String args[])
    {  int n, i, fact=1;
       Scanner scan= new Scanner (System.in);
       System.out.print("Please Enter a No.");
       n=scan.nextInt();
       for (i=n;i>=1;i—)
          {
          fact =fact*i ;
           }
       System.out.println("Factorial of " + n + " is  " +
fact);
}
}
```

Output:

```
C:\javaexprg>javac Factorial.java

C:\javaexprg>java Factorial
Please Enter a No.5
Factorial of 5 is  120
```

10. Write a program to add two matrices.

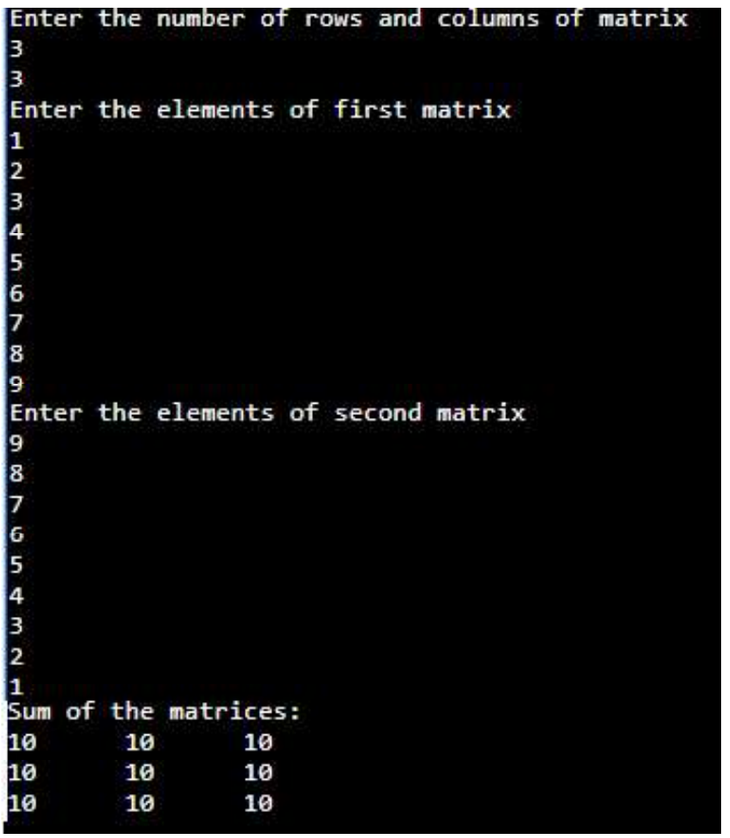
```
import Java.util.Scanner;

class AddTwoMatrix
{
   public static void main(String args[])
    {
       int m, n, c, d;
      Scanner in = new Scanner (System.in);
```

NOTES

```
System.out.println ("Enter the number of rows and columns
of matrix");
      m = in.nextInt ();
n =in.nextInt ();

      int first[][] = new int[m][n];
      int second[][] = new int[m][n];
      int sum[][] = new int[m][n];

System.out.println ("Enter the elements of first matrix");
      for (c = 0; c < m; c++)
         for (d = 0; d < n; d++)
           first[c][d] = in.nextInt();
System.out.println ("Enter the elements of second matrix");

      for (c = 0 ; c < m ; c++)
         for (d = 0 ; d < n ; d++)
           second[c][d] = in.nextInt();

      for (c = 0; c < m; c++)
         for (d = 0; d < n; d++)
            sum[c][d] = first[c][d] + second[c][d];
//replace '+' with '-' to subtract matrices

System.out.println ("Sum of the matrices:");

      for (c = 0; c < m; c++)
      {
         for (d = 0; d < n; d++)
System.out.print(sum[c][d]+"\t");

System.out.println();
      }
   }
}
```

NOTES

Output:



11. Write a program to subtract two matrices.

```
import Java.util.Scanner;

public class MatrixSubtraction
 {

   public static void main(String[] args)
 {
      Scanner s = new Scanner (System.in);
System.out.println ("Enter the number of rows");
      int rows = s.nextInt();
```

NOTES

```
System.out.println ("Enter the number of columns");
      int columns = s.nextInt();
      int matrix1[][] = new int[rows][columns];
      int matrix2[][] = new int[rows][columns];
      int sub[][] = new int[rows][columns];

System.out.println ("Enter the elements of first matrix
:");

      for (int i = 0; i< rows; i++) {
          for (int j = 0; j < columns; j++) {
               matrix1[i][j] = s.nextInt();
           }
       }

System.out.println ("Enter the elements of second matrix
:");

      for (int i = 0; i< rows; i++) {
          for (int j = 0; j < columns; j++) {
               matrix2 [i][j] = s.nextInt();
           }
       }

      for (int i = 0; i< rows; i++) {
          for (int j = 0; j < columns; j++) {
            sub [i][j] = matrix1[i][j] - matrix2[i][j];
           }
       }

System.out.println ("The subtraction of the two matrices
is :");
```

NOTES

```
        for (int i = 0; i< rows; i++) {
             for (int j = 0; j < columns; j++) {
System.out.print("\t" + sub[i][j]);
            }
System.out.println ();
       }
s.close ();
    }
}
```

Output:

```
Enter the number of rows and columns of matrix
3
3
Enter the elements of first matrix
1
2
3
4
5
6
7
8
9
Enter the elements of second matrix
9
8
7
6
5
4
3
2
1
The subtraction of the two matrices is :
        -8      -6      -4
        -2      0       2
        4       6       8
```

12. Write a program to multiply two matrices.

NOTES

```
import Java.util.Scanner;

public class MatrixMultiplication
{
  public static void main(String args[])
   {
      int m, n, p, q, sum = 0, i, j, k;

     Scanner in = new Scanner (System.in);
System.out.println ("Enter the number of rows and columns
of first matrix");
     m = in.nextInt ();
     n = in.nextInt ();

     int first[][] = new int[m][n];

System.out.println ("Enter elements of first matrix");

     for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
          first[i][j] = in.nextInt();

System.out.println ("Enter the number of rows and columns
of second matrix");
     p = in.nextInt ();
     q = in.nextInt ();

     if (n != p)
System.out.println ("The matrices can't be multiplied
with each other.");
     else
     {
```

NOTES

```
        int second[][] = new int[p][q];
        int multiply[][] = new int[m][q];

System.out.println ("Enter elements of second matrix");

        for (i = 0; i < p; i++)
           for (j = 0; j < q; j++)
              second[i][j] = in.nextInt();

        for (i = 0; i < m; i++)
        {
           for (j = 0; j < q; j++)
           {
              for (k = 0; k < p; k++)
              {
                sum = sum + first[i][k]*second[k][j];
              }
              multiply [i][j] = sum;
              sum = 0;
           }
        }

System.out.println ("Product of the matrices :");

        for (i = 0; i < m; i++)
        {
           for (j = 0; j < q; j++)
System.out.print (multiply[i][j]+"\t");

System.out.print ("\n");
        }
      }
   }
}
```

Output:

```
Enter the number of rows and columns of first matrix
3
3
Enter elements of first matrix
1
2
3
4
5
6
7
8
9
Enter the number of rows and columns of second matrix
3
3
Enter elements of second matrix
9
8
7
6
5
4
3
2
1
Product of the matrices:
30  24  18
84  69  54
138 114 90
```

NOTES

13. Write a Java program to demonstrate the concept of inter thread communication.

```
public class ThreadDemo2 extends Thread
{
 public void run()
{
for(int i=0;i<20;i++)
 {
 System.out.println(getName()+":"+i);
 }
 }
 public static void main(String[] args)
 {
 System.out.println("main started");
ThreadDemo2 td=new ThreadDemo2();
ThreadDemo2 td1=new ThreadDemo2();
td.setName("Thread1");
 td1.setName("Thread2");
```

NOTES

```
td.start();
 td1.start();
 td.yield();
 System.out.println("Main Exited");
 }
 }
```

Output:

```
C:\javaprg>java ThreadDemo2
main started
Main Exited
Thread1:0
Thread2:0
Thread1:1
Thread2:1
Thread1:2
Thread2:2
Thread1:3
Thread2:3
Thread1:4
Thread2:4
Thread1:5
Thread2:5
Thread1:6
Thread2:6
Thread1:7
Thread2:7
Thread2:8
Thread2:9
Thread2:10
Thread2:11
Thread2:12
Thread2:13
Thread2:14
Thread2:15
Thread2:16
Thread2:17
Thread2:18
Thread2:19
Thread1:8
Thread1:9
Thread1:10
Thread1:11
Thread1:12
Thread1:13
Thread1:14
Thread1:15
Thread1:16
Thread1:17
Thread1:18
Thread1:19
```

Try Yourself

1. Write a program to print sum of diagonal values of a square matrix.
2. Write a program to find largest and smallest element in a matrix.
3. Write a Java program that searches a value in an m x n matrix.
4. Write a program to calculate area of a circle, a rectangle or a triangle depending on input using overloaded calculate function.

14. Write a Java program to implement the SQL login ID commands using JDBC.

NOTES

```
package Javaapplication5;
import Java.sql.*;
import Java.awt.*;
import Javax.swing.*;
public class NewJFrame extends Javax.swing.JFrame
{
    public NewJFrame() {
        initComponents();
     }
    private void initComponents() {
        jTextField1 = new Javax.swing.JTextField();
        jTextField2 = new Javax.swing.JTextField();
        jButton1 = new Javax.swing.JButton();
        jLabel1 = new Javax.swing.JLabel();
        jLabel2 = new Javax.swing.JLabel();
           setDefaultCloseOperation(Javax.swing.Window
Constants.EXIT_ON_CLOSE);
        jButton1.setText("Login");
                jButton1.addActionListener(new
Java.awt.event.ActionListener()
{
 public void actionPerformed(Java.awt.event.ActionEvent
evt)

{
               jButton1ActionPerformed(evt);
            }
        });
private  void  jButton1ActionPerformed(Java.awt.
event.ActionEvent evt)
{
 try
    {
 //       Connection conn;
```

NOTES

```
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
         String url = "Jdbc:Odbc:g2";
       Connection conn = DriverManager.getConnection(url);
            ResultSet rs;
          Statement stmt=conn.createStatement();
              rs=stmt.executeQuery("select * from mytab");


      while(rs.next())
       {
if(((jTextField1.getText()).equals(rs.getString(1)))&&
((jTextField2.getText()).equals(rs.getString(2)))){
        JOptionPane.showMessageDialog(this, "login
successfull");
     System.exit(0);
}}
              JOptionPane.showMessageDialog(this, "login
unsuccessfull");
               System.exit(0);
           conn.close();
    }
      catch (Exception e) {
             System.err.println("Got an exception! ");
             System.err.println(e.getMessage());
           }
       }
}
```

Output:

15. Write a Java program to implement the SQL commands using JDBC.

NOTES

```
package opertiondemo;
import Java.sql.*;
import Javax.swing.*;
public class operationdemo1 extends Javax.swing.JFrame {
 ResultSet rs1;
    /** Creates new form operationdemo1 */
    public operationdemo1() {
        initComponents();

         try
     {
     Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
       String url = "Jdbc:Odbc:g4";
     Connection conn = DriverManager.getConnection(url);

      Statement     stmt=conn.createStatement
(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_
UPDATABLE);

         rs1=stmt.executeQuery("select * from student");

       }
    catch (Exception e) {
           System.err.println("Got an exception! ");
           System.err.println(e.getMessage());
         }
         }
private  void  jButton1ActionPerformed(Java.awt.event.
ActionEvent evt)
{
    try
{System.out.println("Outside rs1");
     if(rs1.next()){
         System.out.println("Inside rs1");
       String r1=rs1.getString(1);
       String n1=rs1.getString(2);
```

NOTES

```
        String a1=rs1.getString(3);
        jTextField1.setText(r1);
        jTextField2.setText(n1);
        jTextField3.setText(a1);
      }
     else
     {
       JOptionPane.showMessageDialog(this, "eND OF THE
RECORD");
     }
      }catch(Exception e)
  {}
   }
private void jButton3ActionPerformed(Java.awt.event.
ActionEvent evt) {
    try
    {
     Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      String url = "Jdbc:Odbc:g4";
    Connection conn = DriverManager.getConnection(url);
     ResultSet rs;
      Statement stmt=conn.createStatement();
          String qry=("Insert into student
values('"+jTextField1.getText()+"','"+jTextField2.getText()+"',
'"+jTextField3.getText()+"')");
        stmt.executeUpdate(qry);
          JOptionPane.showMessageDialog(this, "Record
inserted");

                  conn.close();
  }
   catch (Exception e)
   {
          System.err.println("Got an exception! ");
          System.err.println(e.getMessage());
       }
 }
```

NOTES

```
private void jButton7ActionPerformed(Java.awt.
event.ActionEvent evt) {
 try
{System.out.println("Outside rs1");
    if(rs1.previous()){
        System.out.println("Inside rs1");
       String r1=rs1.getString(1);
       String n1=rs1.getString(2);
       String a1=rs1.getString(3);
       jTextField1.setText(r1);
       jTextField2.setText(n1);
       jTextField3.setText(a1);
     }
  }catch(Exception e)
  {}
}
private void jButton5ActionPerformed(Java.awt.
event.ActionEvent evt) {
// TODO add your handling code here:
     try
    {
 //      Connection conn;
     Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      String url = "Jdbc:Odbc:g4";
    Connection conn = DriverManager.getConnection(url);
         ResultSet rs;
  Statement stmt=conn.createStatement(ResultSet.
TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
         rs=stmt.executeQuery("select * from student");
        rs.first();
       String r=rs.getString(1);
       String n=rs.getString(2);
       String a=rs.getString(3);
       jTextField1.setText(r);
       jTextField2.setText(n);
       jTextField3.setText(a);
```

NOTES

```
        conn.close();
    }
      catch (Exception e) {
             System.err.println("Got an exception! ");
             System.err.println(e.getMessage());
          }

}
private  void  jButton6ActionPerformed(Java.awt.event.
ActionEvent evt) {
// TODO add your handling code here:

     try
    {
 //       Connection conn;
     Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      String url = "Jdbc:Odbc:g4";
    Connection conn = DriverManager.getConnection(url);

   ResultSet rs;

Statement  stmt=conn.createStatement(ResultSet.
TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
         rs=stmt.executeQuery("select * from student");

              rs.last();
      String r=rs.getString(1);
      String n=rs.getString(2);
      String a=rs.getString(3);
      jTextField1.setText(r);
      jTextField2.setText(n);
      jTextField3.setText(a);
       conn.close();
    }
      catch (Exception e) {
             System.err.println("Got an exception! ");
```

```
            System.err.println(e.getMessage());
        }

}

private void jButton4ActionPerformed(Java.awt.event.
ActionEvent evt) {
// TODO add your handling code here:

     try
    {
 //       Connection conn;
     Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      String url = "Jdbc:Odbc:g4";
    Connection conn = DriverManager.getConnection(url);

   ResultSet rs;

String qry="update student set rollnum="+jTextField1.
getText()+",age="+jTextField3.getText()+"  where
name='"+jTextField2.getText()+"'";
  Statement stmt=conn.createStatement();
     stmt.executeUpdate(qry);
          JOptionPane.showMessageDialog(this, "Record
Updated");

                  conn.close();
  }
    catch (Exception e) {
           System.err.println("Got an exception! ");
           System.err.println(e.getMessage());
        }

}

private void jButton2ActionPerformed(Java.awt.
event.ActionEvent evt)
{
```

NOTES

NOTES

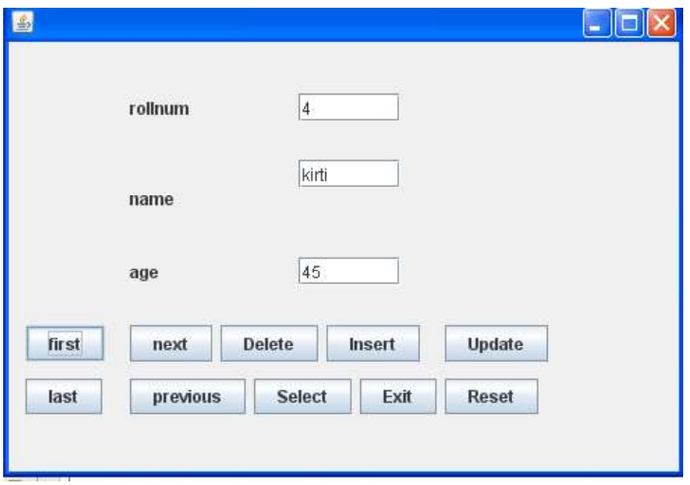
```
// TODO add your handling code here:

     try
    {
 //       Connection conn;
     Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      String url = "Jdbc:Odbc:g4";
    Connection conn = DriverManager.getConnection(url);

   Statement stmt=conn.createStatement();
        stmt.executeUpdate("delete * from student where
rollnum="+jTextField1.getText()+"");
          JOptionPane.showMessageDialog(this, "Record
deleted");

       jTextField1.setText(" ");
       jTextField2.setText(" ");
       jTextField3.setText(" ");
        conn.close();
  }
    catch (Exception e) {
           System.err.println("Got an exception! ");
           System.err.println(e.getMessage());
         }

}

private  void  jButton8ActionPerformed(Java.awt.event.
ActionEvent evt)
{
System.exit(0);
    // TODO add your handling code here:
}

private  void  jButton9ActionPerformed(Java.awt.event.
ActionEvent evt)
{
```

NOTES

```
jTextField1.setText(" ");
       jTextField2.setText(" ");
       jTextField3.setText(" ");
    // TODO add your handling code here:
}

private void jButton10ActionPerformed(Java.awt.event.
ActionEvent evt)
{
// TODO add your handling code here:

     try
    {
 //       Connection conn;
     Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      String url = "Jdbc:Odbc:g4";
    Connection conn = DriverManager.getConnection(url);

    ResultSet rs;

   Statement stmt=conn.createStatement();
  rs=stmt.executeQuery("Select * from student");
        while(rs.next())
{
       String r=rs.getString(1);
       String n=rs.getString(2);
       String a=rs.getString(3);
       jTextField1.setText(r);
       jTextField2.setText(n);
       jTextField3.setText(a);
             }
                   conn.close();
  }
   catch (Exception e) {
          System.err.println("Got an exception! ");
          System.err.println(e.getMessage());
       }}}
```

NOTES

Output:



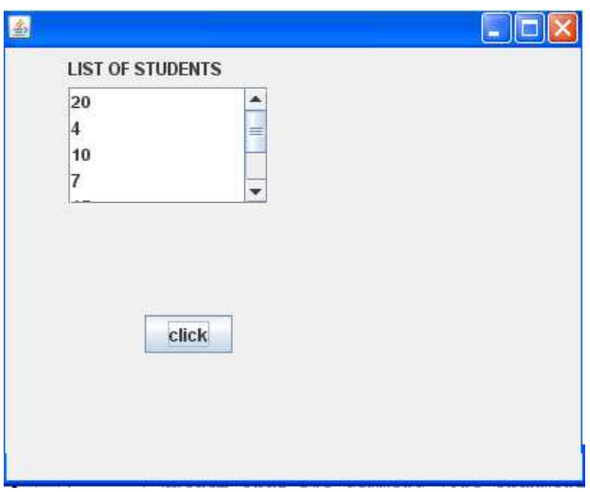
16. Write a Java program to implement the List.

```
package list;

import Java.sql.*;
import Javax.swing.*;
public class NewJFrame extends Javax.swing.JFrame {
    public NewJFrame() {
        initComponents();
     }

   private void jButton1ActionPerformed(Java.awt.event.
ActionEventevt)
{//GEN FIRST:event_jButton1ActionPerformed
DefaultListModel model=new DefaultListModel();
String username=" ";
String password=" ";
        try
        {
         Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
           String url="jdbc:odbc:student";
Connection con=DriverManager.getConnection (url,username,
password);
           Statement st=con.createStatement();
```

NOTES

```
            ResultSet rs=st.executeQuery("Select * from
Table1");
             while(rs.next())
              {
                    model.addElement(rs.getString(1)+  "
"+rs.getString(2)+" "+rs.getString(3));
                 // TODO add your handling code here:
    }//GEN-LAST:event_jButton1ActionPerformed
   jList1.setModel(model);
         }
        catch(Exception e)
         {
            System.out.println(e);
         }
     }
}
```

Output:



17. Write a program to illustrate the use of JDBC connection.

```
package jdbc;
import Java.sql.Connection;
import Java.sql.DriverManager;
import Java.sql.SQLException;
import Java.sql.SQLWarning;
import Java.sql.Statement;
```

NOTES

```
public class JDBC
 {

    /**
     * @param args the command line arguments
     */

    // JDBC database URL
      static final String DB_URL = "jdbc:mysql://
localhost:3306/demo";

    //  Database credentials
    static final String USER = "root";
    static final String PASS = "";

   public static void main(String[] args)
 {
    Connection conn = null;
    Statement stmt = null;
SQLWarning warn = null;

try
{
       //STEP 1: Register JDBC driver
Class.forName("com.mysql.jdbc.Driver");

       //STEP 2: Open a connection
System.out.println("Connecting to a selected database...");
      conn = DriverManager.getConnection(DB_URL, USER,
PASS);
       warn = conn.getWarnings();
System.out.println("Connected database successfully...");

       //STEP 3: Execute a query for table creation
System.out.println("Creating table in given database...");
stmt = conn.createStatement();
```

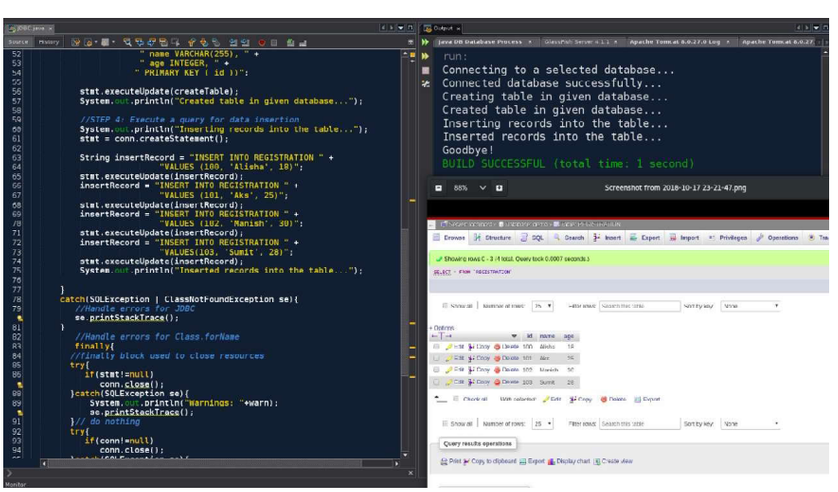
NOTES

```
    String createTable = "CREATE TABLE REGISTRATION" +
                         "(id INTEGER not NULL, " +
                         " name VARCHAR(255), " +
                         " age INTEGER, " +
                        " PRIMARY KEY ( id ))";

stmt.executeUpdate(createTable);
System.out.println("Created table in given database...");

       //STEP 4: Execute a query for data insertion
System.out.println("Inserting records into the table...");
stmt = conn.createStatement();

       String insertRecord = "INSERT INTO REGISTRATION
" +  "VALUES (100, 'Alisha', 18)";
stmt.executeUpdate(insertRecord);
insertRecord = "INSERT INTO REGISTRATION " +  "VALUES
(101, 'Aks', 25)";
stmt.executeUpdate(insertRecord);
insertRecord = "INSERT INTO REGISTRATION " + "VALUES
(102, 'Manish', 30)";
stmt.executeUpdate(insertRecord);
insertRecord = "INSERT INTO REGISTRATION " + "VALUES(103,
'Sumit', 28)";
stmt.executeUpdate(insertRecord);
System.out.println("Inserted records into the table...");

    }
catch(SQLException | ClassNotFoundException se)
{
      //Handle errors for JDBC
se.printStackTrace ();
    }
      //Handle errors for Class.forName
Finally
{
```

NOTES

```
      //finally block used to close resources
try
{
        if (stmt!=null)
conn.close();
}
catch (SQLException se)
{
System.out.println ("Warnings: "+warn);
se.printStackTrace ();
      }// do nothing
try
{
        if (conn!=null)
conn.close();
}
catch(SQLException se)
{
se.printStackTrace();
      }//end finally try
   }//end try
System.out.println("Goodbye!");
}//end main
}//end
```

Output:



18. Write a program to demonstrate the concept of SQL exception, SQL warning.

NOTES

```
package jdbc;
import Java.sql.Connection;
import Java.sql.DriverManager;
import Java.sql.SQLException;
import Java.sql.SQLWarning;
import Java.sql.Statement;

public class JDBC

{
    // JDBC database URL
       static final String DB_URL = "jdbc:mysql://
localhost:3306/demo";
    //  Database credentials
    static final String USER = "root";
    static final String PASS = "";

   public static void main(String[] args)
{
    Connection conn = null;
    Statement stmt = null;
SQLWarning warn = null;

try
{
       //STEP 1: Register JDBC driver
Class.forName("com.mysql.jdbc.Driver");

       //STEP 2: Open a connection
System.out.println("Connecting to a selected database...");
      conn = DriverManager.getConnection(DB_URL, USER,
PASS);
       warn = conn.getWarnings();
System.out.println("Connected database successfully...");
```

NOTES

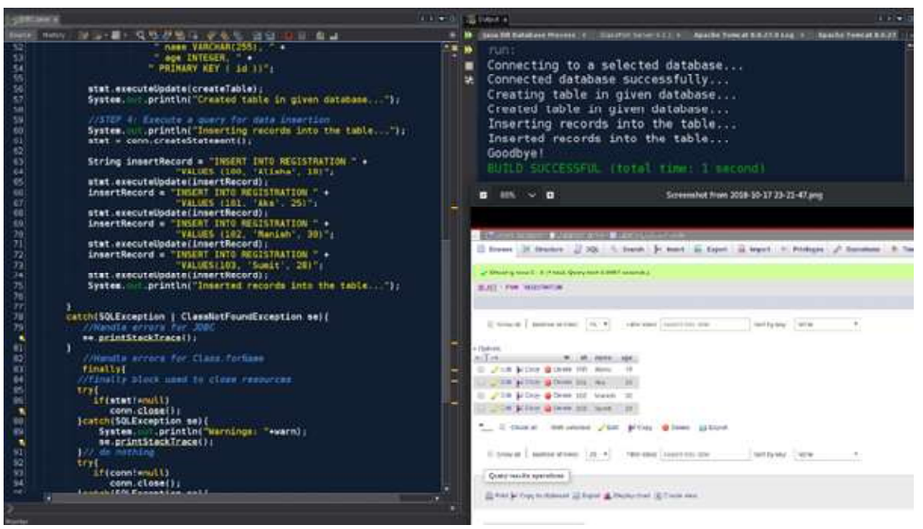
```
        //STEP 3: Execute a query for table creation
System.out.println("Creating table in given database...");
stmt = conn.createStatement();

    String createTable = "CREATE TABLE REGISTRATION " +
                         "(id INTEGER not NULL, " +
                         " name VARCHAR(255), " +
                         " age INTEGER, " +
                        " PRIMARY KEY ( id ))";

stmt.executeUpdate(createTable);
System.out.println("Created table in given database...");

        //STEP 4: Execute a query for data insertion
System.out.println("Inserting records into the table...");
stmt = conn.createStatement();

        String insertRecord = "INSERT INTO REGISTRATION
" + "VALUES (100, 'Alisha', 18)";
stmt.executeUpdate(insertRecord);
insertRecord = "INSERT INTO REGISTRATION " + "VALUES
(101, 'Aks', 25)";
stmt.executeUpdate(insertRecord);
insertRecord = "INSERT INTO REGISTRATION " + "VALUES
(102, 'Manish', 30)";
stmt.executeUpdate(insertRecord);
insertRecord = "INSERT INTO REGISTRATION " + "VALUES(103,
'Sumit', 28)";
stmt.executeUpdate(insertRecord);
System.out.println("Inserted records into the table...");

    }
catch(SQLException | ClassNotFoundException se)
{
```

```
      //Handle errors for JDBC
se.printStackTrace();
    }
      //Handle errors for Class.forName
finally
{
     //finally block used to close resources
try
{
        if(stmt!=null)
conn.close();
}
catch(SQLException se)
{
System.out.println("Warnings: "+warn);
se.printStackTrace();
     }// do nothing
try
{
        if(conn!=null)
conn.close();
}
catch(SQLException se)
{
se.printStackTrace();
     }//end finally try
  }//end try
System.out.println("Goodbye!");
}//end main
}//end JDBCExample
```

NOTES

NOTES

Output:



19. Write a program using TCP/IP client sockets and TCP/IP server sockets.

Client Program

```
import Java.io.DataInputStream;
import Java.io.DataOutputStream;
import Java.io.IOException;
import Java.io.InputStream;
import Java.io.OutputStream;
import Java.net.Socket;

public class Client
{
//main
    public static void main(String [] args)
{
      String serverName = args[0];
     int port = Integer.parseInt(args[1]);
try
{
System.out.println("Connecting to " + serverName + " on
```

```
port " + port);
          try (Socket client = new Socket(serverName,
port))
 {
System.out.println("Just connected to " + client.getRemote
SocketAddress());
OutputStreamoutToServer = client.getOutputStream();
DataOutputStream out = new DataOutputStream(outToServer);

out.writeUTF("Hello from " + client.getLocalSocket
Address());
InputStreaminFromServer = client.getInputStream();
DataInputStream in = new DataInputStream(inFromServer);

System.out.println("Server says " + in.readUTF());
          }
      }
catch (IOException e)
 {
e.printStackTrace();
      }
   }

}
```

NOTES

Server Program

```
import Java.io.DataInputStream;
import Java.io.DataOutputStream;
import Java.io.IOException;
import Java.net.ServerSocket;
import Java.net.Socket;
import Java.net.SocketTimeoutException;
import Java.util.Scanner;

public class Server extends Thread
{
```

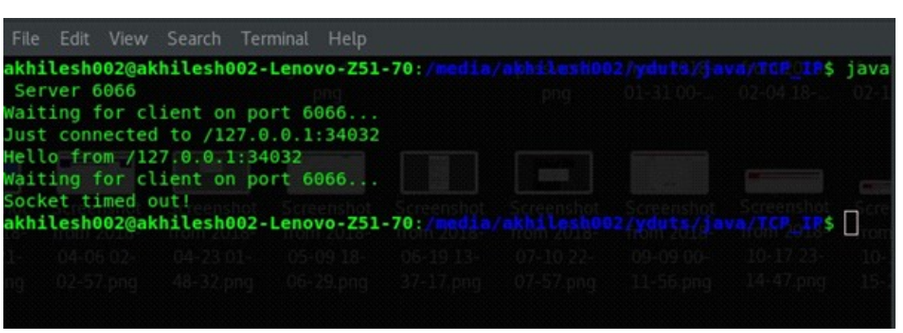
NOTES

```
    private final ServerSocketserverSocket;

  public Server(int port) throws IOException
 {
serverSocket = new ServerSocket(port);
serverSocket.setSoTimeout(10000);
   }

   public void run()
{
     while(true)
{
 try
 {
System.out.println("Waiting for client on port " +
serverSocket.getLocalPort() + "...");
          Socket server = serverSocket.accept();

System.out.println("Just connected to " +
server.getRemoteSocketAddress());
DataInputStream in = new DataInputStream
(server.getInputStream());

System.out.println(in.readUTF());
DataOutputStream out = new DataOutputStream
(server.getOutputStream());
out.writeUTF("Thank you for connecting to " +
server.getLocalSocketAddress()
             + "\nGoodbye!");
server.close();

         }
 catch (SocketTimeoutException s)
{
System.out.println("Socket timed out!");
            break;
         }
```

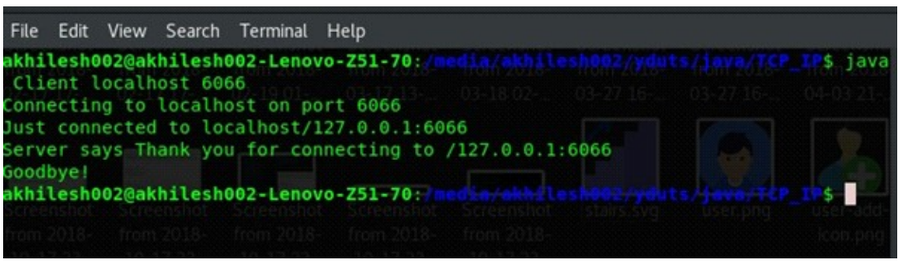
```
catch (IOException e)
{
e.printStackTrace();
            break;
          }
       }
    }

   public static void main(String [] args)
{
      int port = Integer.parseInt(args[0]);
       try
 {
         Thread t = new Server(port);
t.start();
       }
 catch (IOException e)
{
e.printStackTrace();
       }
    }

}
```

NOTES

Output:



NOTES



20. Write a program to illustrate the Client/Server applications using RMI.

Client Program

```
import Java.rmi.registry.LocateRegistry;
import Java.rmi.registry.Registry;

public class Client
 {
   private Client()
{}
   public static void main(String[] args)
 {
      try
 {
        // Getting the registry
 Registry registry = LocateRegistry.getRegistry(null);

       // Looking up the registry for the remote object
        Hello stub = (Hello) registry.lookup("Hello");

        // Calling the remote method using the obtained
object
stub.printMsg();

       // System.out.println("Remote method invoked");
      }
 catch (Exception e)

{
System.err.println("Client exception: " + e.toString());
```

```
e.printStackTrace();
      }
   }
}
```

NOTES

Server Program

```
import Java.rmi.registry.Registry;
import Java.rmi.registry.LocateRegistry;
import Java.rmi.RemoteException;
import Java.rmi.server.UnicastRemoteObject;

public class Server extends ImplExample
{
   public Server()
 {}
  public static void main(String args[])
{
try
{
        // Instantiating the implementation class
ImplExampleobj = new ImplExample();

       // Exporting the object of implementation class
        // (here we are exporting the remote object to
the stub)
Hello stub = (Hello) UnicastRemoteObject.exportObject(obj,
0);

          // Binding the remote object (stub) in the
registry
Registry registry = LocateRegistry.getRegistry();

registry.bind("Hello", stub);
System.err.println("Server ready");
      }
catch (Exception e)
{
```

NOTES

```
System.err.println("Server exception: " + e.toString());
e.printStackTrace();
      }
   }
}
```

- **Hello Interface**

```
import Java.rmi.Remote;
import Java.rmi.RemoteException;

// Creating Remote interface for our application
public interface Hello extends Remote
{
  void printMsg() throws RemoteException;
}
```

- **Implementation Example**

```
// Implementing the remote interface
public class ImplExample implements Hello
 {

   // Implementing the interface method
   public void printMsg()
{
System.out.println("This is an example RMI program");
   }
}
```

21. Write a Java program to implement the Remote Method Invocation. Interface

```
import Java.rmi.*;
interface Bank extends Remote
{
double getAmount(double p,double t) throws RemoteException;
}
```

NOTES

Bank Server

```
import Java.rmi.*;
import Java.rmi.server;
public class BankImpl extends UnicastRemoteObject
implements Bank
{
public BankImpl throws RemoteException
{


}
double getAmount(double p,double t) throws RemoteException
{
return p*Math.pow(1.41,t);
}
}
```

RMI Registry

```
import Java.rmi.*;
import Javax.naming.*;
public class BankServer
{
public static void main(String args[])
{
BankImpl  centralbank=new BankImpl();
Naming.rebind("uti",centralbank);
}
}
```

Bank Client

```
import Java.rmi.*;
import Javax.naming.*;
public class Bankclient
{
public static void main(String args[]) throws
RemoteException
```

NOTES

```
{
String url="rmi://localHost//uti";
Bank b=(Bank) Naming.lookup(url);
System.out.println(b.getAmount(4000,3));
}
}
```

22. Write a simple programs using Bean Development Kit and JAR files.

Main Class

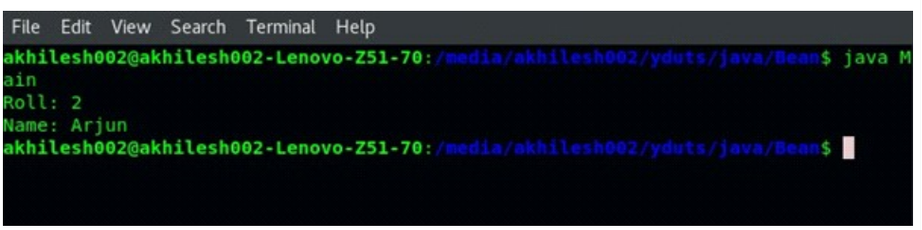
```
public class Main
{
public static void main(String args[])
{

Student s=new Student();//object is created

//setting value to the object
s.setRoll(2);
s.setName("Arjun");

//getting value from the object
System.out.println(s.getRoll());
System.out.println(s.getName());
}
}
```

Java Class

```
public class Student implements Java.io.Serializable
{
private int roll;
private String name;

public Student()
{}
```

```
public void setRoll(int roll)
{
this.roll=roll;
}

public int getRoll()
{
return roll;
}

public void setName(String name)
{
this.name=name;
}

public String getName()
{
return name;
}

}
```

Output:



Design Patterns

Design Patterns are very popular among software developers. A design pattern is a well described solution to a common software problem. Design Patterns are already defined and provides industry standard approach to solve a recurring problem, so it saves time if we sensibly use the design pattern. There are many Java design patterns that we can use in our Java based projects. It leads to faster development since design patterns are already defined and debug.

NOTES

There are two Categories of design patterns:

1. Core Java (or JSE) design patterns
2. JEE design patterns

These categories are further divided in subcategories:

1. Core Java design patterns
   - Creational design pattern
   - Structural design pattern
   - Behavioral design pattern
2. JEE design patterns

23. Write a program to demonstrate events and methods.

```
import Javax.swing.*;
import Java.awt.event.*;
class TestEvent extends JFrame implements ActionListener
{
JTextFieldtf;
TestEvent()
{
  //create components
JFramefrm = new JFrame();
tf=new JTextField();
tf.setBounds(60,50,170,20);
frm.add(tf);
JButton b=new JButton("click me");
b.setBounds(100,120,120,30);
frm.add(b);
//register listener
b.addActionListener(this);//passing current instance
  //add components and set size, layout and visibility
frm.setSize(300,300);
```

```
frm.setLayout(null);
frm.setVisible(true);
frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public void actionPerformed(ActionEvent e)
{
greet();
}

public void greet()
{
tf.setText("Hello ");
}
public static void main(String args[])
{
new TestEvent();
}
}
```

Output:

NOTES

NOTES

24. Write a program to create a servlet to read the parameters.

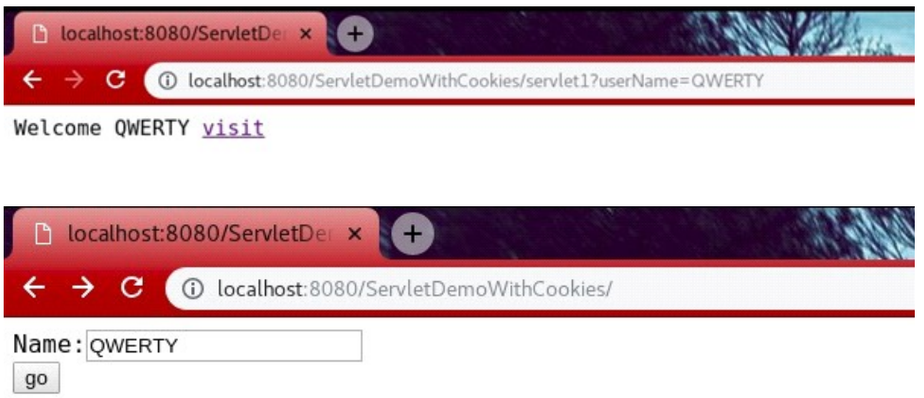
```
// Import required Java libraries
import Java.io.*;
import Javax.servlet.*;
import Javax.servlet.http.*;

// Extend HttpServlet class
public class HelloForm extends HttpServlet
 {

   // Method to handle GET method request.
     public  void  doGet(HttpServletRequest  request,
HttpServletResponse response)
     throws ServletException, IOException
 {

      // Set response content type
response.setContentType("text/html");

PrintWriter out = response.getWriter();
     String title = "Using GET Method to Read Form Data";
     String docType =  "<!doctype html public \"-//w3c/
/dtd html 4.0 " + "transitional//en\">\n";
```

NOTES

```
out.println(docType +      "<html>\n" +     "<head><title>"
            + title + "</title></head>\n" +
            "<body bgcolor = \"#f0f0f0\">\n" +      "<h1
            align = \"center\">" + title + "</h1>\n" +
            "<ul>\n" +
            "  <li><b>First Name</b>: "
            + request.getParameter("first_name") +
             "\n" + "  <li><b>Last Name</b>: "
            + request.getParameter("last_name") + "\n" +
             "</ul>\n" +
             "</body>\n"+
            "</html>"
        );
    }

   // Method to handle POST method request.
     public  void  doPost(HttpServletRequest  request,
HttpServletResponse response)
     throws ServletException, IOException
 {

doGet(request, response);
    }
}
```

Output:

NOTES



25. Write a program to demonstrate the use of servlet.

```
import Java.io.*;
import Javax.servlet.http.*;
public class FirstServlet extends HttpServlet
{
public void doGet(HttpServletRequest request,
HttpServletResponse response)
{
try
{

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String username = request.getParameter("userName");
        out.print("Welcome " + username);

        Cookie cookie = new Cookie("user", username);
        response.addCookie(cookie);

        HttpSession session=request.getSession();
        session.setAttribute("uname",username);
```

```
        out.print("\n<a href='servlet2'>visit</a>");

        out.close();

}
catch(IOException e)
{
System.out.println(e);
}
    }

}

// Program:
import Java.io.*;
import Javax.servlet.http.*;

public class SecondServlet extends HttpServlet
{
public void doGet(HttpServletRequest request,
HttpServletResponse response)
{
try
{
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

out.println("Content from cookies.....");
               Cookie ck[]=request.getCookies();
                for (Cookie ck1 : ck)
{
out.print("<br>" + ck1.getName() + " " + ck1.getValue());
//printing name and value of cookie
                }
```

NOTES

NOTES

```
out.println("<br><br>————————————————————————————<br>");
out.println("Content from session......<br>");
HttpSession session=request.getSession(false);
String n=(String)session.getAttribute("uname");
out.print("Hello "+n);
       out.close();

}
catch(IOException e)
{
System.out.println(e);
}
    }
}
```

26. Write a servlet program to set some cookies, send it to browser, print cookie information and send it as HTML response.

```
package com.journaldev.servlet.cookie;
import Java.io.IOException;
import Java.io.PrintWriter;
import Javax.servlet.ServletException;
import Javax.servlet.annotation.WebServlet;
import Javax.servlet.http.Cookie;
import Javax.servlet.http.HttpServlet;
import Javax.servlet.http.HttpServletRequest;
import Javax.servlet.http.HttpServletResponse;

@WebServlet("/cookie/SetCookie")
public class SetCookie extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static int count = 0;

    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException {
```

NOTES

```
        PrintWriter out = response.getWriter();
        Cookie[] requestCookies = request.getCookies();

        out.write("<html><head></head><body>");
        out.write("<h3>Hello Browser!!</h3>");

        if(requestCookies != null){
        out.write("<h3>Request Cookies:</h3>");
        for(Cookie c : requestCookies){
           out.write("Name="+c.getName()+",
Value="+c.getValue()+", Comment="+c.getComment()
                +",   Domain="+c.getDomain()+",
MaxAge="+c.getMaxAge()+", Path="+c.getPath()
                +", Version="+c.getVersion());
           out.write("<br>");
        }
        }
        //Set cookies for counter, accessible to only this
servlet
        count++;
        Cookie counterCookie = new Cookie("Counter",
String.valueOf(count));
        //add some description to be viewed in browser
cookie viewer
        counterCookie.setComment("SetCookie Counter");
        //setting max age to be 1 day
        counterCookie.setMaxAge(24*60*60);
        //set path to make it accessible to only this
servlet
        counterCookie.setPath("/ServletCookie/cookie/
SetCookie");

        //adding cookie to the response
        response.addCookie(counterCookie);

        //set a domain specific cookie
        Cookie domainCookie = new Cookie("Test", "Test
Cookie"+String.valueOf(count));
```

NOTES

```
            domainCookie.setComment("Test Cookie");
            response.addCookie(domainCookie);

            out.write("</body></html>");
        }

    }
```

27. Write a Java program for session tracking to find out the creation time and the last accessed time for a session using the HttpSession object.

In the following Java program the HttpSession object is used for finding out the time of session creation and also the time when the session was last accessed. The program will create a new session with the request if the session does not exist.

```
// Import required java libraries
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

// Extend HttpServlet class
public class SessionTrack extends HttpServlet {

     public void doGet(HttpServletRequest request,
HttpServletResponse response)
      throws ServletException, IOException {

        // Create a session object if it is already not
created.
      HttpSession session = request.getSession(true);

      // Get session creation time.
                    Date createTime = new
                Date(session.getCreationTime());

      // Get last access time of this web page.
                    Date lastAccessTime = new
                Date(session.getLastAccessedTime());

      String title = "Welcome Back to My Website";
      Integer visitCount = new Integer(0);
      String visitCountKey = new String("visitCount");
      String userIDKey = new String("userID");
      String userID = new String("ABCD");

      // Check if this is new comer on your web page.
      if (session.isNew()) {
         title = "Welcome to My Website";
         session.setAttribute(userIDKey, userID);
      } else {
            visitCount = (Integer)session.getAttribute
                (visitCountKey);
            visitCount = visitCount + 1;
            userID = (String)session.getAttribute
                (userIDKey);
      }
```

NOTES

```
        session.setAttribute(visitCountKey,  visitCount);
        // Set response content type
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String docType =
          "<!doctype html public \"-//w3c//dtd html 4.0 " +
           "transitional//en\">\n";
        out.println(docType +
           "<html>\n" +
           "<head><title>" + title + "</title></head>\n" +
              "<body bgcolor = \"#f0f0f0\">\n" +
                   "<h1 align = \"center\">" + title + "</
h1>\n" +
           "<h2 align = \"center\">Session
                    Information</h2>\n" +
           "<table border = \"1\" align = \"center\">\n" +
               "<tr bgcolor = \"#949494\">\n" +
                "  <th>Session info</th><th>value</th>
                </tr>\n" +
                "<tr>\n" +
                "  <td>id</td>\n" +
                 "  <td>" + session.getId() + "</td>
                 </tr>\n" +
                 "<tr>\n" +
                 "  <td>Creation Time</td>\n" +
                 "  <td>" + createTime + "  </td>
                 </tr>\n" +
                 "<tr>\n" +
                 "  <td>Time of Last Access</td>\n" +
                  "  <td>" + lastAccessTime + "  </td>
                  </tr>\n" +
                  "<tr>\n" +
                  "  <td>User ID</td>\n" +
                  "  <td>" + userID + "  </td>
                  </tr>\n" +
                  "<tr>\n" +
                  "  <td>Number of visits</td>\n" +
                  "  <td>" + visitCount + "</td>
                  </tr>\n" +
               "</table>\n" +
            "</body>
          </html>"
        );
    }
}
```

Now compile the above Java program servlet **`SessionTrack`** by creating the appropriate entry in the web.xml file.

NOTES

Run the Java program at http://localhost:8080/SessionTrack which will display the following result, when for the first time the program is run.

Output:

Welcome to My Website
Session Information

Session info	**value**
ID	0AE3EC93FF44E3C525B4351B77ABB2D5
Creation Time	Tue Jun 08 17:26:40 GMT+04:00 2010
Time of Last Access	Tue Jun 08 17:26:40 GMT+04:00 2010
User ID	ABCD
Number of visits	0

When you run the same servlet program for the second time, it would display the following result.

Output:

Welcome Back to My Website
Session Information

Info type	**Value**
ID	0AE3EC93FF44E3C525B4351B77ABB2D5
Creation Time	Tue Jun 08 17:26:40 GMT+04:00 2010
Time of Last Access	Tue Jun 08 17:26:40 GMT+04:00 2010
User ID	ABCD
Number of visits	1

28. Write a Java program using Session Tracking Servlet on how to track a session in servlet using the HttpSession object.

Session tracking helps to identify the client which was interacting with the server and was idle for some time is the same client or the other when it tries to interact next time to the server.

Session Management

```
package roseindia.webContext;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.Date;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```

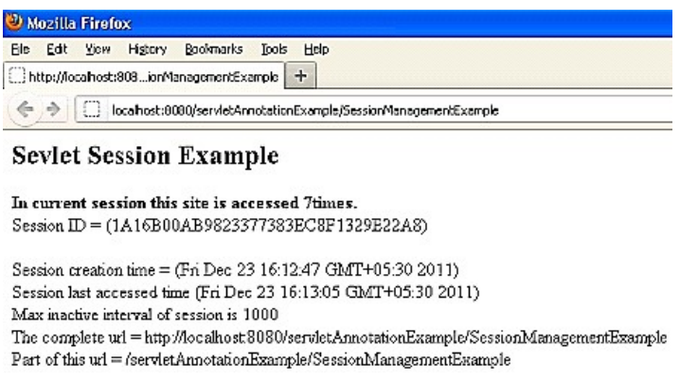
NOTES

```
@WebServlet("/SessionManagementExample")
public class SessionManagementExample extends HttpServlet
{
private static final long serialVersionUID = 1L;
public void doGet(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException
{
response.setContentType("text/html");
PrintWriter out = response.getWriter();
HttpSession session = request.getSession();
Date creationTime= new Date(session.getCreationTime());
Date lastAccessedTime= new
 Date(session.getLastAccessedTime());
session.setMaxInactiveInterval(1000);
Integer count;
count = (Integer)session.getAttribute("Count");
if (count == null)
{
count = 0;
}
else
{
count = new Integer(count + 1);
}
session.setAttribute("Count", count);
try
{
out.println("<h2>Sevlet Session Example</h2>");
if(count==0 || count==1)
{
out.println("<b>In current session this site is accessed "
+ count + " time.</b>");
}
else
out.println("<b>In current session this site is accessed
"+ count + "times. </b>");
out.println("<br>Session ID = (" + session.getId() + ")</
br>");
out.println("<br>Session creation time =
("+creationTime+")");
out.println("<br>Session last accessed time
("+lastAccessedTime+")");
out.println("<br>Max inactive interval of session is
"+session.getMaxInactiveInterval());
out.println("<br>The complete url =
"+request.getRequestURL());
out.println("<br>Part of this url =
"+request.getRequestURI());
}
catch(Exception ex)
{
```

NOTES

```
out.println(ex);
}
}
public void doPost(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException
{
doGet(request,response);
}
}
```

Output :



29. Write a program for creation of button using JApplet.

```
//Create AWT Button Example
//This Java example shows how to create a Button using AWT
Button class.

import Java.applet.Applet;
import Java.awt.Button;

/*
<applet code="CreateAWTButtonExample" width=200 height=200>
</applet>
*/

public class CreateAWTButtonExample extends Applet
{
    public void init()
{
// To create a button use Button () constructor.
Button button1 = new Button ();
// Set button caption or label using void setLabel(String
text)  method of AWT Button class
    button1.setLabel ("My Button 1");

//  To create button with the caption use  Button(String
text) constructor of  AWT Button class.
        Button button2 = new Button ("My Button 2");
       //add buttons using add method
       add(button1);
```

```
        add(button2);
    }
}
```

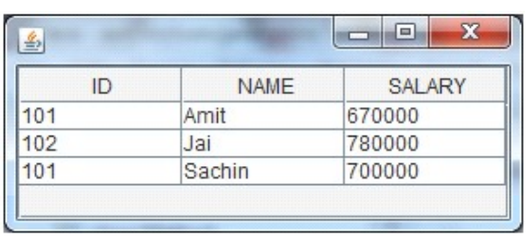
Output:

NOTES

30. Write a program to create table using Java Applet.

```
import Javax.swing.*;
public class TableExample
 {
JFrame f;
TableExample()
{
    f=new JFrame();
    String data[][]={ {"101","Amit","670000"},
                      {"102","Jai","780000"},
                      {"101","Sachin","700000"}};
    String column[]={"ID","NAME","SALARY"};
JTablejt=new JTable(data,column);
jt.setBounds(30,40,200,300);
JScrollPanesp=new JScrollPane(jt);
f.add(sp);
f.setSize(300,400);
f.setVisible(true);
 }
public static void main(String[] args)
{
    new TableExample();
}
}
```

Output:



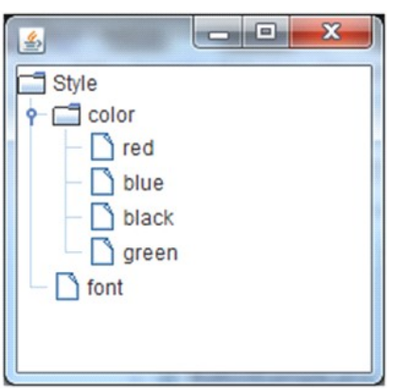
ID	NAME	SALARY
101	Amit	670000
102	Jai	780000
101	Sachin	700000

NOTES

31. Write a program for creating a tree using JApplet.

```
import Javax.swing.*;
import Javax.swing.tree.DefaultMutableTreeNode;
public class TreeExample
{
JFrame f;
TreeExample(){
    f=new JFrame();
DefaultMutableTreeNode style=new DefaultMutableTreeNode
("Style");
DefaultMutableTreeNodecolor=new DefaultMutableTreeNode
("color");
DefaultMutableTreeNode font=new DefaultMutableTreeNode
("font");
style.add(color);
style.add(font);
DefaultMutableTreeNode red=new DefaultMutableTreeNode
("red");
DefaultMutableTreeNode blue=new DefaultMutableTreeNode
("blue");
DefaultMutableTreeNode black=new DefaultMutableTreeNode
("black");
DefaultMutableTreeNode green=new DefaultMutableTreeNode
("green");
color.add(red); color.add(blue); color.add(black);
color.add(green);
JTreejt=new JTree(style);
f.add(jt);
f.setSize(200,200);
f.setVisible(true);

}
public static void main(String[] args)
{
    new TreeExample();
}
}
```

Output:

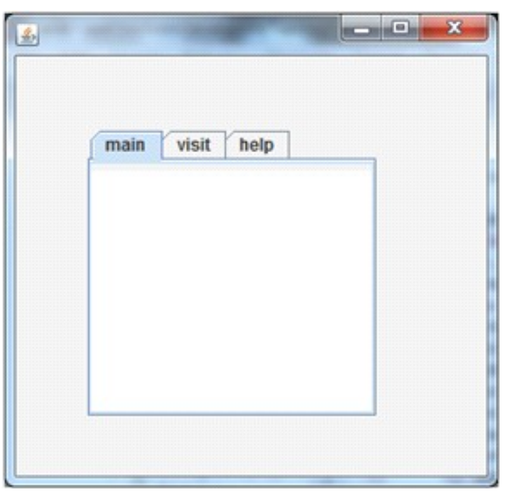


NOTES

32. Write a program to create panes in Java Applet.

```
import Javax.swing.*;
public class TabbedPaneExample
 {
JFrame f;
TabbedPaneExample()
{
    f=new JFrame();
JTextArea ta=new JTextArea(200,200);
JPanel p1=new JPanel();
    p1.add(ta);
JPanel p2=new JPanel();
JPanel p3=new JPanel();
JTabbedPanetp=new JTabbedPane();
tp.setBounds(50,50,200,200);
tp.add("main",p1);
tp.add("visit",p2);
tp.add("help",p3);
f.add(tp);
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
}
public static void main(String[] args)
 {
    new TabbedPaneExample();
}
}
```

Output:



33. Write a Java program for sum of two numbers using Applet.

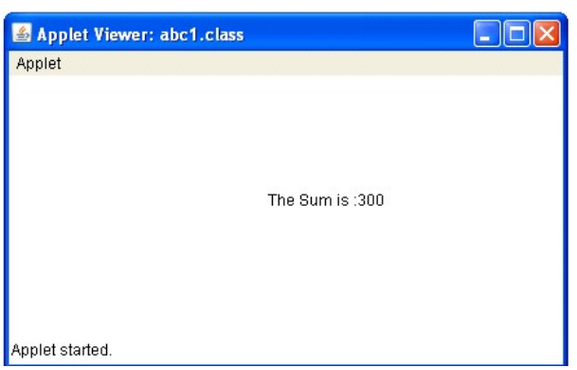
```
import Java.awt.*;
import Java.applet.*;
/*Coding of HTML File <applet code = abc1.class width= 200
```

NOTES

```
height=200> </applet> */
public class abc1 extends Applet
{
public void paint(Graphics g)
 {
int a=100;
int b=200;
int sum = a+b;
String s = "The Sum is :" + String.valueOf(sum);
g.drawString( s, 200,100);
} }
```

Output:

```
C:\javaprg>javac SwingMenu.java

C:\javaprg>java SwingMenu

C:\javaprg>javac abc1.java

C:\javaprg>appletviewer abc1.java

C:\javaprg>appletviewer abc1.java
```

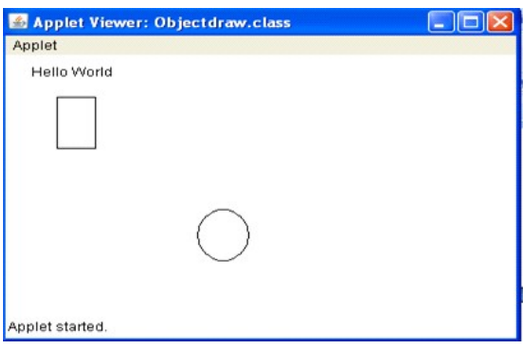


34. Write a Java Program for Applet using drawString (), drawRect () and drawOval ().

```
 import Java.awt.*;
import Java.applet.*;
/*<applet code= "Objectdraw.class" height=400 width=400>
</applet>*/
public class Objectdraw extends Applet
{
 public void paint(Graphics g)
{
g.drawString("Hello World",20,20);
 g.drawRect(40,40,30,50);
g.drawOval(150,150,40,50);
} }
```

Output:

```
C:\javaprg>javac Objectdraw.java
C:\javaprg>appletviewer Objectdraw.java
```



NOTES

35. Write a Java program to create a banner using Applet.

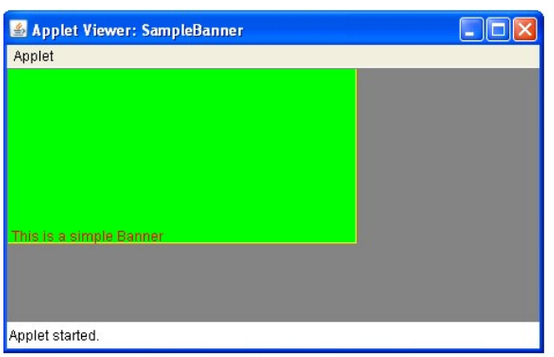
```
import Java.awt.*;
import  Java.applet.*;
 /*<HTML> <BODY> <APPLET CODE = "SampleBanner" WIDTH =
"460" HEIGHT = "220"></APPLET> </BODY> </HTML> */
public class SampleBanner extends Applet implements Runnable
{
 String str = "This is a simple Banner developed by Class
Naren ";
Thread t ;
boolean b;
public void init()
{
setBackground(Color.gray);
 setForeground(Color.yellow);
}
public void start()
{
t = new Thread(this);
b = false; t.start();
}
public void run ()
{
char ch;
 for( ; ; )
{
try
 {
repaint();
Thread.sleep(250);
ch = str.charAt(0);
 str = str.substring(1, str.length());
 str = str + ch;
}
```

NOTES

```
catch(InterruptedException e)
{}
}
} public void paint(Graphics g)
{
g.drawRect(1,1,300,150);
 g.setColor(Color.green);
g.fillRect (1,1,300,150);
g.setColor(Color.red);
 g.drawString(str, 1, 150);
} }
```

Output:

```
C:\javaprg>javac SampleBanner.java
C:\javaprg>appletviewer SampleBanner.java
```



36. Write a Java program for bouncing a ball using Applet.

```
 import Java.applet.Applet;
import Java.awt.Color;
import Java.awt.Graphics;
/*<applet code= "Bounce.class" height=900 width=900> </
applet>*/
 class Ball
{
int x,y,radius,dx,dy;
Color BallColor;
 public Ball(int x,int y,int radius,int dx,int dy,Color
bColor)
 {
this.x=x;
this.y=y;
this.radius=radius;
this.dx=dx;
this.dy=dy;
BallColor=bColor;
} }
 public class Bounce extends Applet implements Runnable
{
```

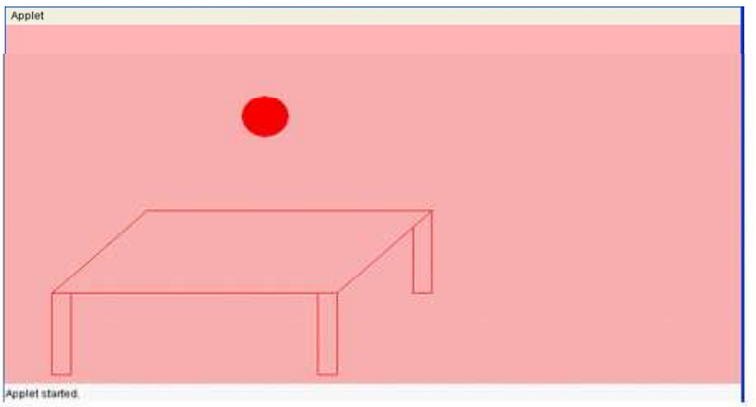
NOTES

```
Ball redBall;
 public void init()
 {
redBall=new Ball(250,80,50,2,4,Color.red);
Thread t=new Thread(this);
 t.start();
}
 public void paint(Graphics g)
{
g.setColor(redBall.BallColor);
 setBackground(Color.pink);
//g.setcolor(redBall.BallColor);
g.fillOval(redBall.x,  redBall.y,  redBall.radius,redBall
.radius);
g.drawLine(150,400,50,500);
g.drawLine(150,400,450,400);
g.drawLine(50,500,350,500);
g.drawLine(450,400,350,500);
g.drawRect(50,500,20,100);
 g.drawRect(330,500,20,100);
g.drawLine(450,400,450,500);
g.drawLine(430,500,450,500);
g.drawLine(430,500,430,420);
}
public void run()
{ while(true)
 {
try
 {
displacementOperation(redBall);
Thread.sleep(20);
repaint();
}
catch(Exception e)
{
}
 }
 }
public void displacementOperation(Ball ball)
{
if(ball.y >= 400 || ball.y <= 0)
{
ball.dy=-ball.dy; } ball.y=ball.y+ball.dy;
 }
}
```

Output:

```
C:\javaprg>javac Bounce.java

C:\javaprg>appletviewer Bounce.java
```

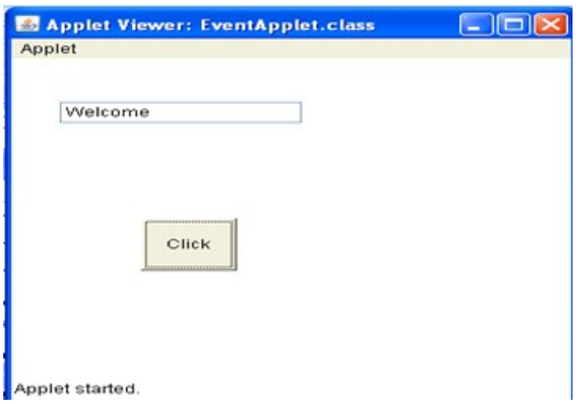
NOTES



37. Write a Java program that prints a message by clicking on the button using AWT Events and Applets.

```
import Java.applet.*;
import Java.awt.*;
import Java.awt.event.*;
public class EventApplet extends Applet implements
ActionListener
{
Button b;
TextField tf;
 public void init()
{
tf=new TextField();
tf.setBounds(30,40,150,20);

b=new Button("Click");
b.setBounds(80,150,60,50);

add(b);add(tf);
b.addActionListener(this);

setLayout(null);
}

 public void actionPerformed(ActionEvent e)
{
  tf.setText("Welcome");
 }
}
/* In the above example, we have created all the controls
in init() method because it is invoked only once.
myapplet.html
1. <html>
2. <body>
3. applet code="EventApplet.class"width="300" height="300">
4.  </applet>
5.  </body>
6.  </html>  */
```

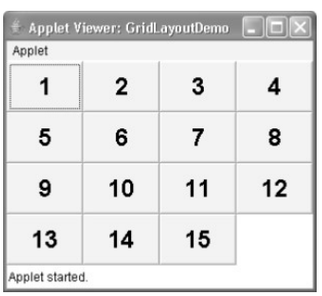
Output:



38. Write a Java program to create a grid layout.

```
import Java.awt.*;
import Java.applet.*;
/*
<applet code="GridLayoutDemo" width=300 height=200>
</applet>
*/
public class GridLayoutDemo extends Applet
{
static final int n = 4;
public void init()
  {
setLayout(new GridLayout(n, n));
setFont(new Font("SansSerif", Font.BOLD, 24));
for(int i = 0; i < n; i++)
{
for(int j = 0; j < n; j++)
   {
int k = i * n + j;
if(k > 0)
add(new Button("" + k));
     }
   }
}
}
```

Output:



NOTES

NOTES

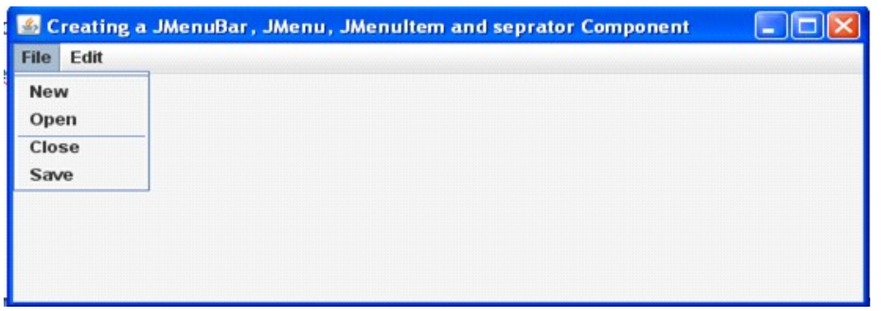
39. Write a Java program to demonstrate an application involving GUI with controls menus and event handling.

```
import Javax.swing.*;
public class SwingMenu
{
public static void main(String[] args)
 {
SwingMenu s = new SwingMenu();
 }
public SwingMenu()
{
JFrame frame = new JFrame("Creating a JMenuBar, JMenu,
JMenuItem and seprator Component");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
JMenuBar menubar = new JMenuBar();
JMenu filemenu = new JMenu("File");
filemenu.add(new JSeparator());
JMenu editmenu = new JMenu("Edit");
editmenu.add(new JSeparator());
JMenuItem fileItem1 = new JMenuItem("New");
JMenuItem fileItem2 = new JMenuItem("Open");
JMenuItem fileItem3 = new JMenuItem("Close");
fileItem3.add (new JSeparator());
JMenuItem fileItem4 = new JMenuItem ("Save");
JMenuItem editItem1 = new JMenuItem ("Cut");
JMenuItem editItem2 = new JMenuItem ("Copy");
editItem2.add (new JSeparator());
JMenuItem editItem3 = new JMenuItem ("Paste");
JMenuItem editItem4 = new JMenuItem ("Insert");
filemenu.add(fileItem1); filemenu.add(fileItem2);
filemenu.add(fileItem3); filemenu.add(fileItem4);
editmenu.add(editItem1);
editmenu.add(editItem2);
editmenu.add(editItem3);
editmenu.add(editItem4);
menubar.add(filemenu);
menubar.add(editmenu);
frame.setJMenuBar(menubar);
frame.setSize(400,400);
frame.setVisible(true);
} }
```

Output:

```
C:\javaprg>javac SwingMenu.java

C:\javaprg>java SwingMenu
```

NOTES



40. Write a program with AWT classes.

```
import Java.awt.*;
import Java.awt.event.*;

public class AWTGraphicsDemo extends Frame
 {
  public AWTGraphicsDemo()
{
super ("Java AWT Examples");
prepareGUI();
   }

   public static void main(String[] args)
{
AWTGraphicsDemoawtGraphicsDemo = new AWTGraphicsDemo();
awtGraphicsDemo.setVisible(true);
   }

   private void prepareGUI()
{
setSize(400,400);
addWindowListener(new WindowAdapter()
 {
public void windowClosing(WindowEventwindowEvent){
System.exit(0);
         }
      });
   }

   @Override
   public void paint(Graphics g)
{
      Graphics2D g1 = (Graphics2D) g;
      Font font1 = new Font ("Serif", Font.PLAIN, 24);
      g1.setFont (font1);
      g1.setColor (Color.BLUE);
     g1.drawString ("Welcome in Java AWT class", 50, 70);
```

NOTES

```
        Graphics2D g2 = (Graphics2D) g;
        Font font2 = new Font ("Times New Roman", 2, 24);
        g2.setFont (font2);
        g2.setColor (Color.BLACK);
       g2.drawString ("Welcome in Java AWT class", 50, 120);
    }
}
```