Lab Manual

Program 1 : Setup Java Programming development environment using: Command prompt.(Classpath and path setup) Any IDE (Eclipse, Netbeans, VScode, Jcreatoretc.).

Soln. :

(a) Path setup

1. After installing jdk. Open command prompt.
2. Set Path using.

```
Set CLASSPATH = C:\jdk\bin
```

(b) Launch

Step 0 : Launch Eclipse

1. Eclipse by running "eclipse.exe" from the Eclipse installed directory.
2. Choose an appropriate directory for your *workspace*, i.e., where you would like to save your files (e.g., c:\myproject\eclipse for Windows).
3. If the "Welcome" screen shows up, close it by clicking the "cross" button next to the "Welcome" title.

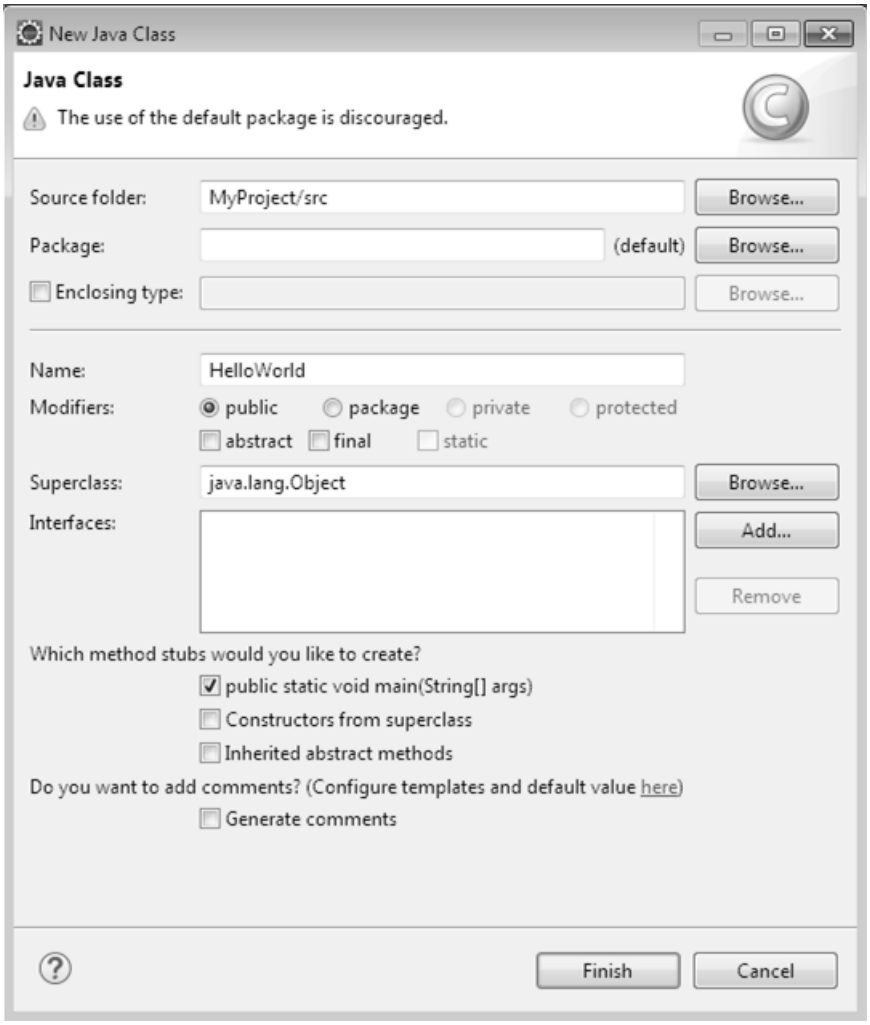
Step 1 : Create a new Java Project

To create a new Java project

1. Select "File" menu ⇒ "New" ⇒ "Java project" (or "File" ⇒ "New" ⇒ "Project" ⇒ "Java project").
2. The "New Java Project" dialog pops up.
3. In "Project name", enter "MyProject".
4. Check "Use default location".
5. In "JRE", select "Use default JRE (currently 'JDK10.0.x')". But make sure that your JDK is 1.8 and above.
6. In "Project Layout", check "Use project folder as root for sources and class files".
   Click "Next" button.
7. Uncheck "Create module-info.java file" box (if it is checked) ⇒ Finish.

Step 2 : Write a Hello-world Java Program

1. In the "Package Explorer" (left side pane) ⇒ Right-click on "MyProject" ⇒ New ⇒ Class.
2. The "New Java Class" dialog pops up.
3. In "Source folder", keep the "MyProject".
4. In "Package", delete the content if it is not empty.
5. In "Name", enter "HelloWorld".
6. Check "public static void main(String ar[ ])".
7. Don't change the rest.



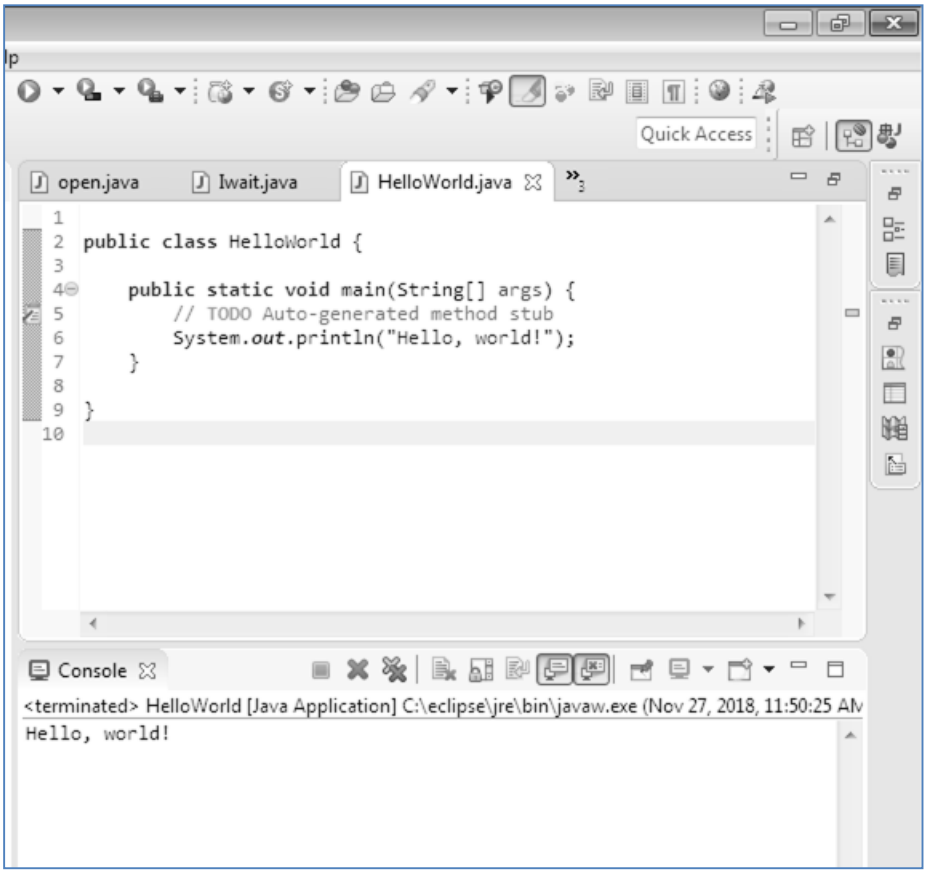
Push "Finish" button.

8. The source file "HelloWorld.java" opens on the editor panel (the center pane). Enter the following codes :

```
public class HelloWorld {
  public static void main(String ar[ ]) {
    System.out.println("Hello, world!");
  }
}
```

Step 3 : Compile and Execute the Java Program

1. There is no need to compile the Java source file in Eclipse explicitly. It is because Eclipse performs the so-called incremental compilation, i.e., the Java statement is compiled as and when it is entered.
2. To run the program, right-click anywhere on the source file "HelloWorld.java" ⇒ Run As ⇒ Java Application.
3. The output "Hello, world!" appears on the Console panel (the bottom pane).



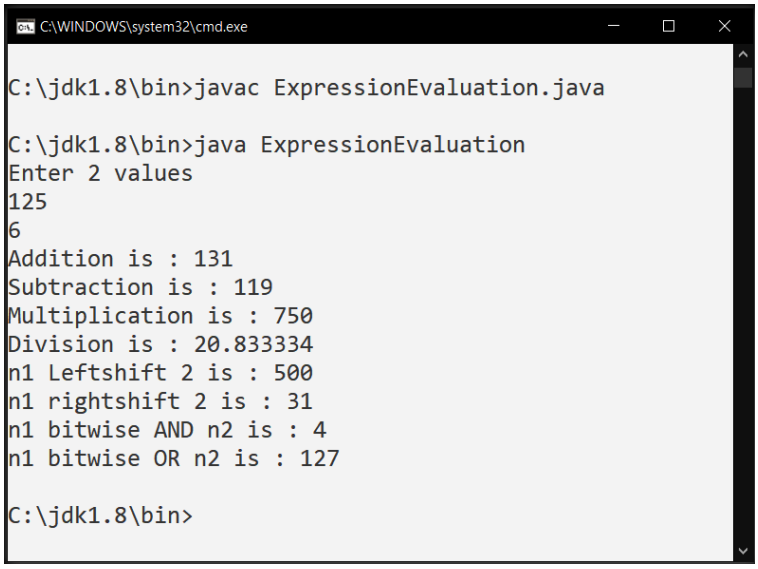
Program 2 : Write programs to evaluate different types of expressions.

Soln. :

```
import java.util.Scanner;
class ExpressionEvaluation
{
  public static void main(String ar[ ])
  {
    int n1, n2, ad, sub, mult;
    int leftShift, rightShift, bitwiseAnd, bitwiseOr;
    float div;
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter 2 values");
    n1 = sc.nextInt( );
    n2 = sc.nextInt( );
    ad = n1 + n2;
    sub = n1 - n2;
    mult = n1 * n2;
    div = n1 / (float)n2;
    System.out.println("Addition is : " + ad);
```

```
    System.out.println("Subtraction is : " + sub);
    System.out.println("Multiplication is : " + mult);
    System.out.println("Division is : " + div);
    leftShift = n1 << 2;
    rightShift = n1 >> 2;
    bitwiseAnd = n1 & n2;
    bitwiseOr = n1 | n2;
    System.out.println("n1 Leftshift 2 is : " + leftShift);
    System.out.println("n1 rightshift 2 is : " + rightShift);
    System.out.println("n1 bitwise AND n2 is : " + bitwiseAnd);
    System.out.println("n1 bitwise OR n2 is : " + bitwiseOr);
  }
}
```

The output of above program will be :



Program 3 : Write programs to demonstrate use of :

if statements (all forms of if statement Switch – Case statement

Different types of Loops(for,while and do..while).

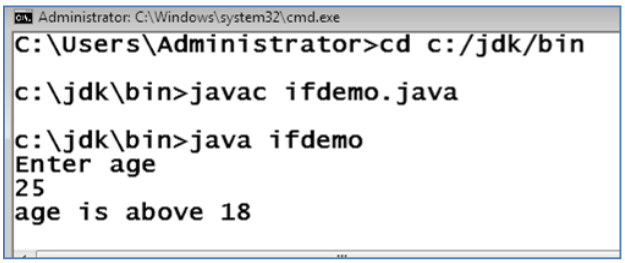
Soln. :

The if statement

```
import java.util.Scanner;
class ifdemo
{
public static void main(String ar[ ])
{
int age;
```

```
Scanner s1=new Scanner(System.in);
System.out.println("Enter age");
age=s1.nextInt( );
if(age>18)
System.out.println("age is above 18");
}
}
```

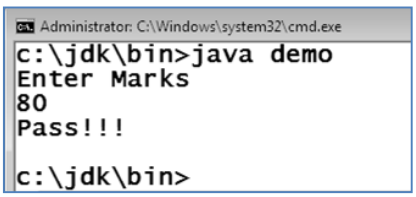
Output



The if else Statement

```
import java.util.Scanner;
class demo
{
public static void main(String ar[ ])
{
int marks;
Scanner s1=new Scanner(System.in);
System.out.println("Enter Marks");
marks=s1.nextInt( );
if(marks>40)
System.out.println("Pass!!!");
else
System.out.println("Fail!!!");
}
}
```

Output



else if Ladder

```
import java.io.*;
class Exbuff
{
public static void main(String ar[ ]) throws IOException
{
int Java,Sys,IP,total,avg;
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
System.out.print("Enter marks of Java: ");
Java = Integer.parseInt(br.readLine( ));
System.out.print("Enter marks of Sys: ");
Sys = Integer.parseInt(br.readLine( ));
System.out.print("Enter marks of IP: ");
IP = Integer.parseInt(br.readLine( ));
total = Java + Sys + IP;
avg = total / 3;
System.out.println("Total marks : "+total);
System.out.println("Average : "+avg);
if(Java>=40 && Sys>=40 && IP>=40)
{
if(avg>=80)
System.out.println("Grade : A");
else if(avg>=60)
System.out.println("Grade : B");
else if(avg>=40)
System.out.println("Grade : C");
}
else
System.out.println("Fail...");
}
}
```

Marks are accepted and converte d in int

Student should pass in all subjects

Output

```
Administrator: C:\Windows\system32\cmd.exe

c:\jdk\bin>java Exbuff
Enter marks of Java: 99
Enter marks of Sys: 98
Enter marks of IP: 23
Total marks : 220
Average : 73
Fail...

c:\jdk\bin>
```

Switch Case statement

```
import java.io.*;
class Exbuff
{
public static void main(String ar[ ]) throws IOException
{
int n1=0,n2=0,r,ch;
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
System.out.println("1 : Addition");
System.out.println("2 : Subtraction");
System.out.println("3 : Multiplcation");
System.out.println("4 : Division");
System.out.print("Select your choice : ");
ch = Integer.parseInt(br.readLine( ));
if(ch>=1 && ch<=4)
{
System.out.print("Enter first number : ");
n1 = Integer.parseInt(br.readLine( ));
System.out.print("Enter second number : ");
n2 = Integer.parseInt(br.readLine( ));
}
switch(ch)
{
case 1:
r = n1 + n2;
System.out.println("Summation is "+r);
break;
case 2:
r = n1 - n2;
System.out.println("Subtraction is "+r);
break;
case 3:
r = n1 * n2;
System.out.println("Multiplication is "+r);
break;
case 4:
```

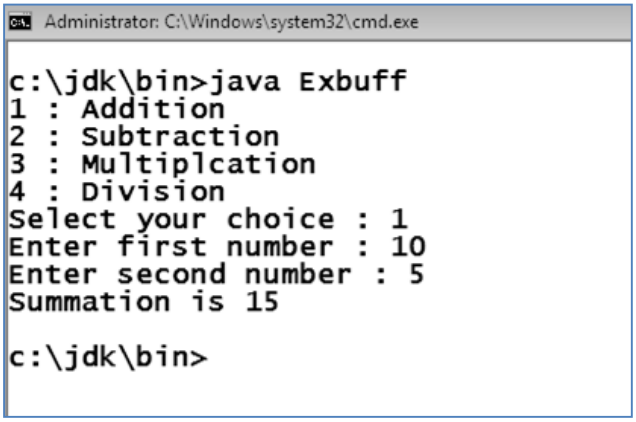
Displaying menu

Accept numbers only for valid choice

```
if(n2!=0)
{
r = n1 / n2;
System.out.println("Division is "+r);
}
else
System.out.println("Cannot divide by zero");
break;
default :
System.out.println("Invalid choice");
}
}
}
```

Division by zero is not allowed

Output

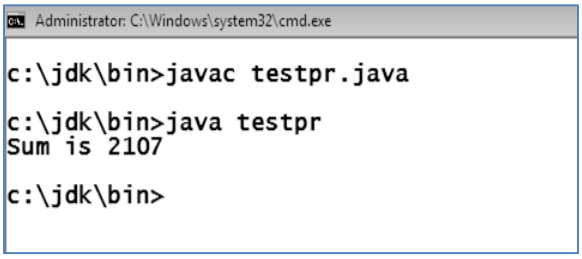


use of Looping Statement 'for

```
class testpr
{
public static void main(String ar[ ])
{
int i,sum;
sum = 0;
for(i=101;i<200;i++)
{
if(i%7==0)
{
 sum = sum + i;
}
}
```

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

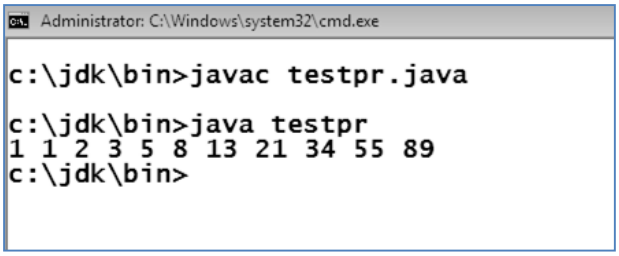
Output



Do -While

```
class testpr
{
public static void main(String ar[ ])
{
int a,b,c,i;
        a = 1;
        b = 2;
        i = 1;
        System.out.print("1 1 2");
        do
        {
              c = a + b;
              System.out.print(" "+c);
              a = b;
              b = c;
              i = i + 1;
        }while(i<9);
}
}
```

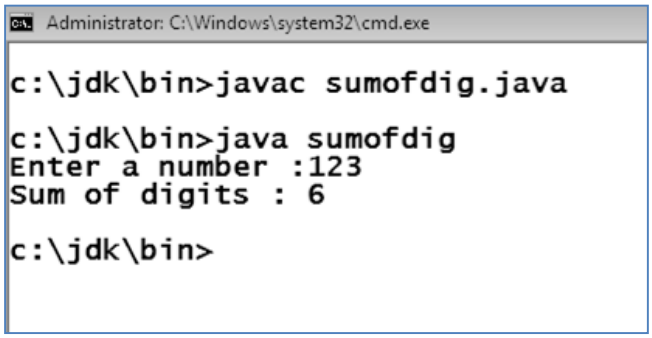
Output



While loop

```
import java.util.*;
class sumofdig
{
public static void main(String ar[ ])
{
Scanner sc = new Scanner(System.in);
int n,rem,sum;
sum = 0;
System.out.print("Enter a number :");
n = sc.nextInt( );
while(n>0)
{
rem = n % 10;
sum = sum + rem ;
n = n / 10;
}
System.out.println("Sum of digits : "+sum);
}
}
```

Output



Program 4 : Write programs for implementation of differentmethods of :
String class.
StringBuffer class.

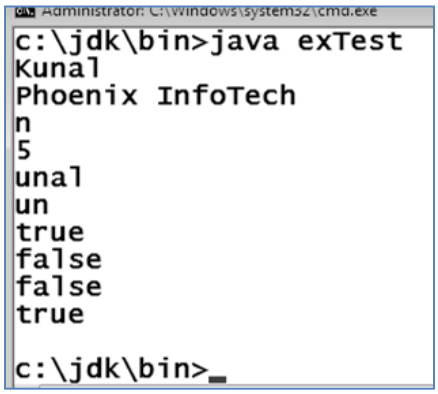
Soln. :

String class

```
class exTest
{
public static void main(String ar[ ])
```

```
{
String str = "Kunal";
String str1 = "Phoenix InfoTech";
System.out.println(str);
System.out.println(str1);
System.out.println(str.charAt(2));
System.out.println(str.length( ));
System.out.println(str.substring(1));
System.out.println(str.substring(1,3));
System.out.println(str.startsWith("Ku"));
System.out.println(str.endsWith("abc"));
System.out.println(str.equals("KUNAL"));
System.out.println(str.equalsIgnoreCase("KUNAL"));
}
}
```

Output

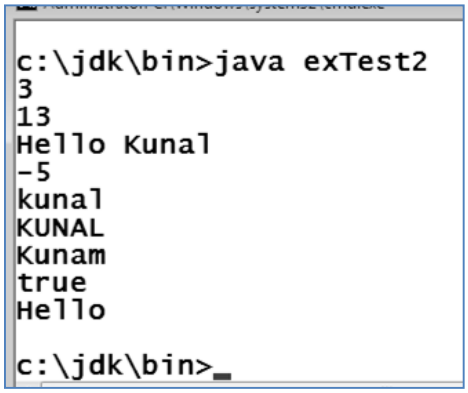


StringBuffer class

```
class exTest2
{
public static void main(String ar[ ])
{
String str = "Kunal";
String str1 = "Phoenix InfoTech";
System.out.println(str1.indexOf('e'));
System.out.println(str1.lastIndexOf('e'));
System.out.println("Hello ".concat(str));
System.out.println(str.compareTo(str1));
System.out.println(str.toLowerCase( ));
```

```
System.out.println(str.toUpperCase( ));
System.out.println(str.replace('l','m'));
System.out.println(str1.contains("Info"));
System.out.println(" Hello ".trim( ));
}
}
```

Output



Program 5 : Write programs to demonstrate :

Use of Array.

Use of Vectors

Soln. :

Use of Array

One dimensional array

```
class Exarray
{
  public static void main (String ar[ ])
  {
   int[] myArray;
   myArray = new int[5];
   myArray[0] = 10;
   myArray[1] = 20;
   myArray[2] = 30;
   myArray[3] = 40;
   myArray[4] = 50;
   for (int i = 0; i < myArray.length; i++)
```

Declares an Array of integers

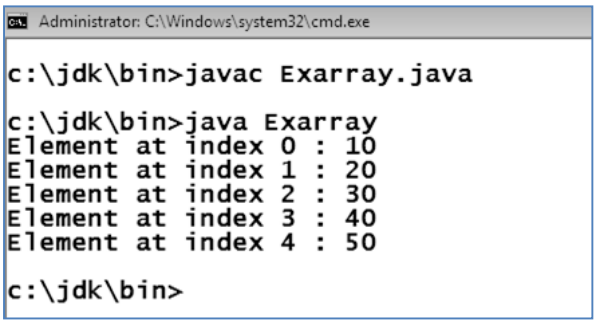
Allocating memory for 5 integers.

Initialize the first elements of the array

```
    System.out.println("Element at index " + i +
                       " : "+ myArray[i]);
  }
}
```

Accessing the elements of the specified array

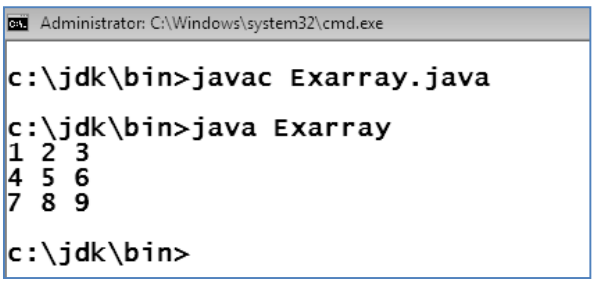
Output



Two dimensional array

```
class Exarray
{
   public static void main (String ar[ ])
   {
   int myarr[][]={{1,2,3},{4,5,6},{7,8,9}};
for(int i=0;i<3;i++)
{
 for(int j=0;j<3;j++)
 {
  System.out.print(myarr[i][j]+" ");
 }
 System.out.println( );
}
   }
}
```

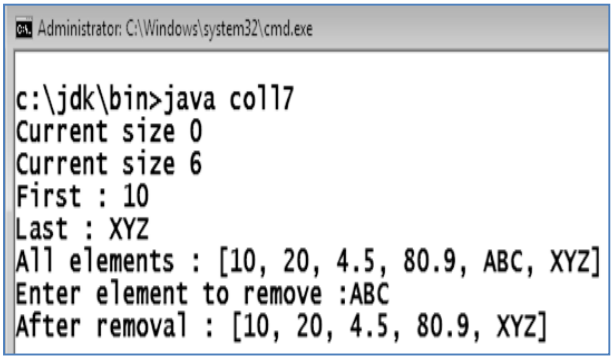
Output



Use of Vectors :

```
import java.util.*;
class coll7
{
public static void main(String ar[ ])
{
Scanner sc = new Scanner(System.in);
Vector v = new Vector(5);
System.out.println("Current size "+v.size( ));

v.addElement(new Integer(10));
v.addElement(new Integer(20));
v.addElement(new Float(4.5));
v.addElement(new Float(80.90));
v.addElement("ABC");
v.addElement("XYZ");

System.out.println("Current size "+v.size( ));
System.out.println("First : "+v.firstElement( ));
System.out.println("Last : "+v.lastElement( ));
System.out.println("All elements : "+v);

System.out.print("Enter element to remove :");
String ele = sc.next( );
v.removeElement(ele);
System.out.println("After removal : "+v);
}
}
```

Output



Program 6 : Write programs using Wrapper Class :

to convert primitive into object.

to convert object into primitive.

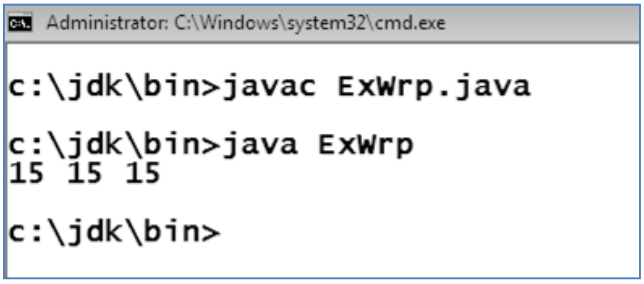
Soln. :

```
public class ExWrp
{
public static void main(String ar[ ]){

int x=15;
Integer y = Integer.valueOf(x);
Integer z = x;
System.out.println(x+" "+y+" "+z);
}
}
```

Converting int to integer

autoboxing, now compiler will write Integer.valueOf(x) internally

Output



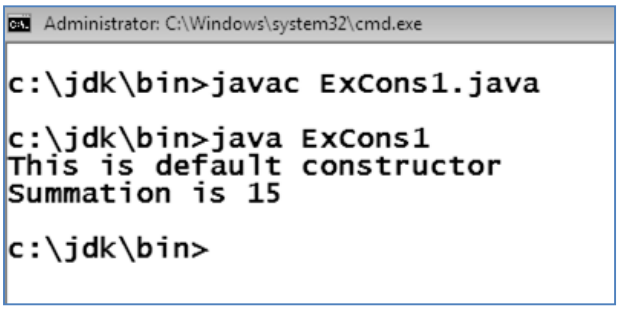
Wrapper Class to convert object into primitive

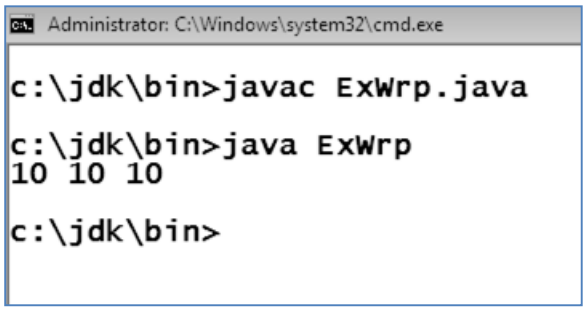
```
public class ExWrp
{
public static void main(String ar[ ]){
Integer x=new Integer(10);

int y=x.intValue( );
int z=x;
System.out.println(x+" "+y+" "+z);
}
}
```

Converting integer to int

Unboxing, now compiler will write x.intValue internally.

Output





Program 7 : Develop a program for implementation of different types of constructors.

Soln. :

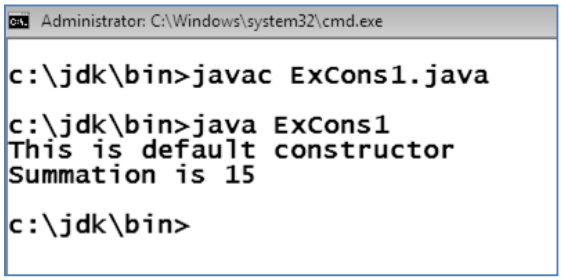
(a) Develop a program for implementation of constructor

```
class ExCons
{
int n1,n2,sum;
ExCons( )
{
System.out.println("This is default constructor");
n1 = 10;
n2 = 5;
}
void cal( )
{
sum = n1 + n2;
System.out.println("Summation is "+sum);
}
}
class ExCons1
{
public static void main(String ar[ ])
{
ExCons obj = new ExCons( );
obj.cal( );
}
}
```

Definition of default constructor

Default constructor get called

Output



(b) Develop a program for implementation of multiple constructors in a class

```
class ExCons
{
int n1,n2,sum;
ExCons( )
{
n1 = 0;
n2 = 0;
}
ExCons(int x, int y)
{
System.out.println("This is parameterized constructor");
n1 = x;
n2 = y;
}
void cal( )
{
sum = n1 + n2;
System.out.println("Summation is "+sum);
}
}
class ExCons1
{
public static void main(String ar[ ])
{
new ExCons( );
ExCons obj = new ExCons(10,20);
obj.cal( );
}
}
```

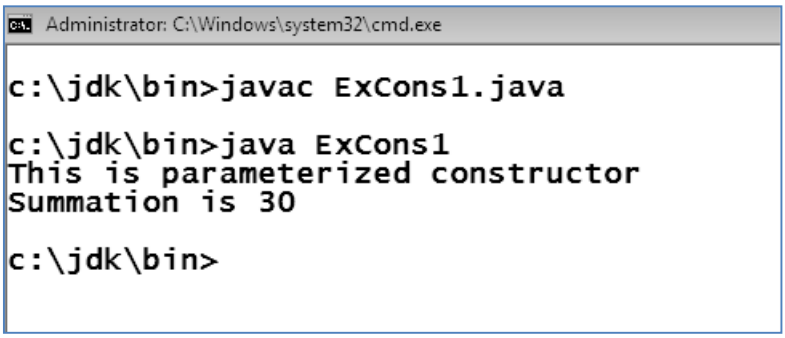
Definition of default constructor

Definition of parameterized

Default constructor get called

Parameterized constructor get called

Output



Program 8 : Develop program to implement : Single inheritance, Multilevel inheritance

Soln. :

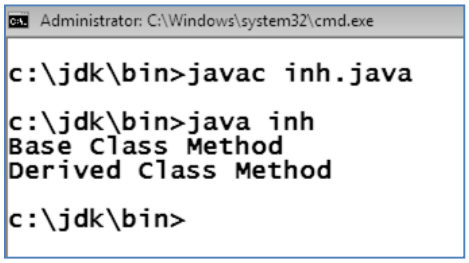
Single Inheritance

```
class A
{
    void displayBase( )
    {
    System.out.println("Base Class Method");
    }
}

class B extends A
{
    void displayDerived( )
    {
    System.out.println("Derived Class Method");
    }
}
class inh
{
    public static void main(String ar[ ])
    {
      B obj = new B( );
      obj.displayBase( );
      obj.displayDerived( );
    }
}
```

Extends keyword is used to inherit a class

Base class method can be accessed using derived class object

Output

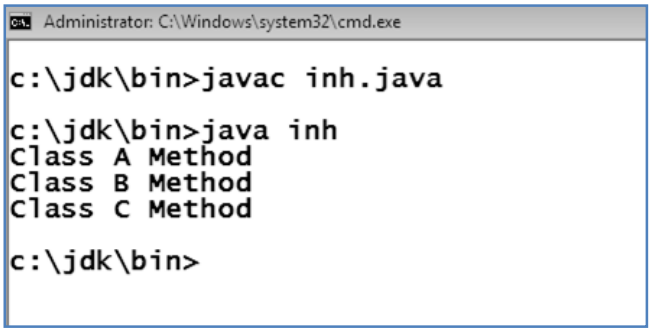


Multilevel Inheritance

```
class A
{
    void display1( )
    {
    System.out.println("Class A Method");
    }
}
class B extends A
{
    void display2( )
    {
    System.out.println("Class B Method");
    }
}
class C extends B
{
    void display3( )
    {
    System.out.println("Class C Method");
    }
}
class inh
{
    public static void main(String ar[ ])
    {
      C obj = new C( );
      obj.display1( );
      obj.display2( );
      obj.display3( );
    }
}
```

Object of class C can access members of both class A and B

Output



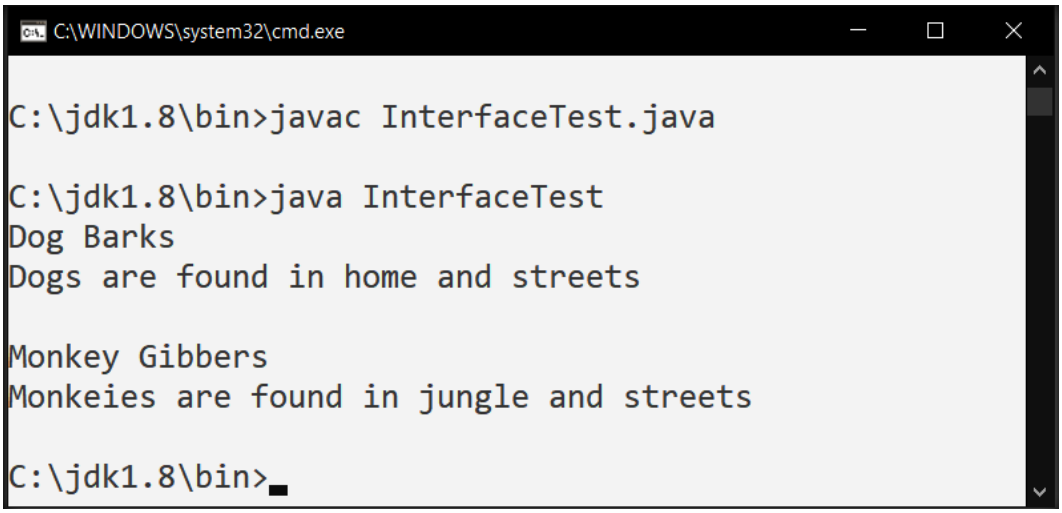
Program 9 : * Develop program for implementation of interface .

Soln. :

```
interface Animal
{
   public void sound( );
   public void foundin( );
}
class Dog implements Animal
{
   public void sound( )
   {
      System.out.println("Dog Barks");
   }

   public void foundin( )
   {
      System.out.println("Dogs are found in home and streets");
   }
}
class Monkey implements Animal
{
   public void sound( )
   {
      System.out.println("Monkey Gibbers");
   }
   public void foundin( )
   {
      System.out.println("Monkeies are found in jungle and streets");
   }
}
```

```
class InterfaceTest
{
  public static void main(String ar[ ])
  {
    Dog d1 = new Dog( );
    d1.sound( );
    d1.foundin( );

    System.out.println( );
    Monkey m1 = new Monkey( );
    m1.sound( );
    m1.foundin( );
  }
}
```

The output of above program will be :



Program 10 : Write programs to demonstrate use of :

Built in packages

User defined packages.

Soln. :

Program 10.1 : Builtinpackages

```
import java.util.Vector;
import java.util.Scanner;
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.io.DataInputStream;
import java.io.IOException;
```

```
class DemoBuiltinPackages
{
  public static void main(String ar[ ]) throws UnknownHostException, IOException
  {
    // Accepting 3 Strings using class Scanner
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter 3 Names");
    String s1 = sc.next( );
    String s2 = sc.next( );
    String s3 = sc.next( );

    // Accepting integer, float and String using DataInputStream
    DataInputStream dis = new DataInputStream(System.in);
      System.out.println("Enter integer, float and String values");
    int n1 = Integer.parseInt(dis.readLine( ));
    float f1 = Float.parseFloat(dis.readLine( ));
    String s4 = dis.readLine( );

    // Obtianing InetAddress of localhost
      System.out.println("Obtaining InetAddress of localhost");
    InetAddress i1 = InetAddress.getLocalHost( );

    // Adding all above objects into Vector
    Vector v1 = new Vector(5);
      System.out.println("Adding elements into Vector");
    v1.addElement(s1);
    v1.addElement(s2);
    v1.addElement(s3);
    v1.addElement(n1);
    v1.addElement(f1);
    v1.addElement(s4);
    v1.addElement(i1);

      System.out.println("All Elements added in Vector");

  }
}
```

The output of above program will be :

Program 10.2 : Userdefinedpackages.

File 1 : ArithmeticOperations.java

```
package mypack1;
public class ArithmeticOperations
{
	public int getAddition(int x, int y)
	{
		return(x+y);
	}
	public int getSubtraction(int x, int y)
	{
		return(x-y);
	}
}
```

File 2 : RelationalOperations.java

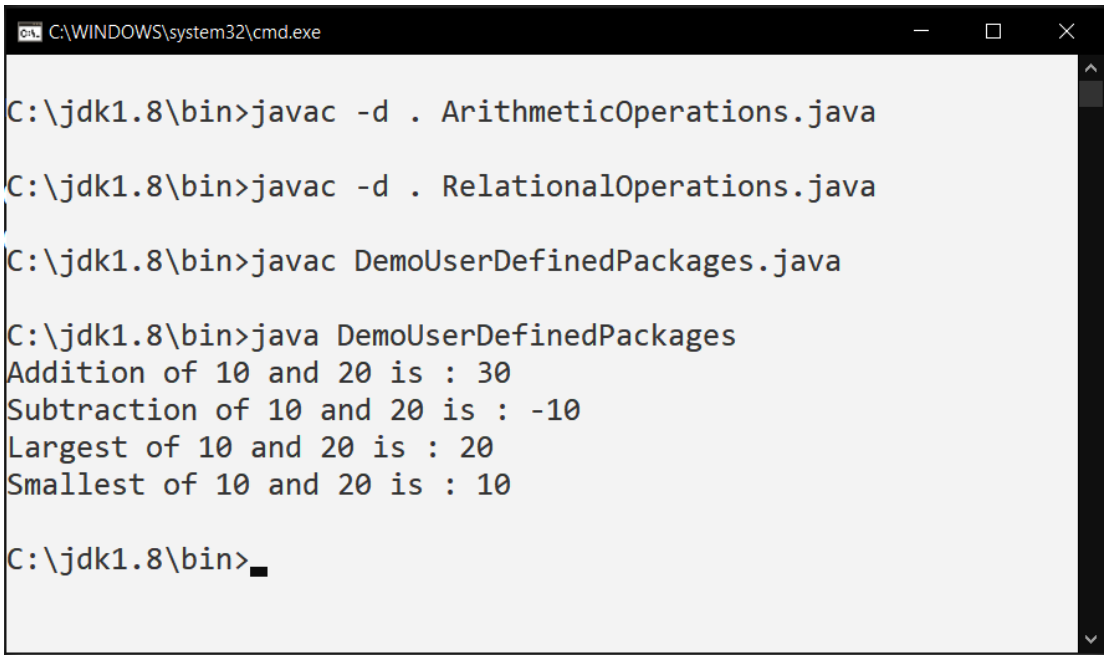
```
package mypack1;
public class RelationalOperations
{
	public int getLargest(int x, int y)
	{
		return x>y ? x : y;
	}
```

```
    public int getSmallest(int x, int y)
    {
        return x<y ? x : y;
    }
}
```

File 3 : DemoUserDefinedPackages.java

```
import mypack1.ArithmeticOperations;
import mypack1.RelationalOperations;
class DemoUserDefinedPackages
{
    public static void main(String ar[ ])
    {
        mypack1.ArithmeticOperations op1 = new mypack1.ArithmeticOperations( );
        mypack1.RelationalOperations op2 = new mypack1.RelationalOperations( );

        System.out.println("Addition of 10 and 20 is : " + op1.getAddition(10,20) );
        System.out.println("Subtraction of 10 and 20 is : " + op1.getSubtraction(10,20) );

        System.out.println("Largest of 10 and 20 is : " + op2.getLargest(10,20) );
        System.out.println("Smallest of 10 and 20 is : " + op2.getSmallest(10,20) );
    }
}
```

The output of above program will be :



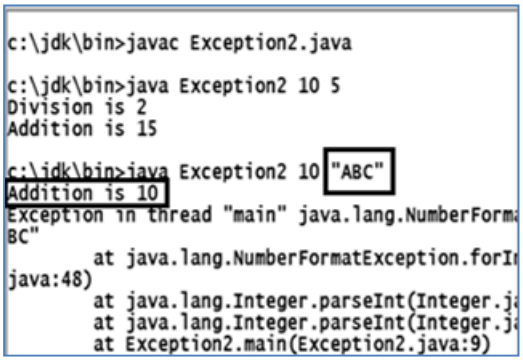
Program 11 : Write programs for implementation of try, catch and finally block.

Soln. :

```
class Exception2
{
 public static void main (String ar[ ])
  {
    int a=0,b=0,res;
   try
   {
      a=Integer.parseInt(args[0]);
      b=Integer.parseInt(args[1]);
     res = a/b;
     System.out.println("Division is "+res);
   }
 catch(ArithmeticException e)
   {
    System.out.println("Can not divide by zero");
   }
 catch(ArrayIndexOutOfBoundsException e)
   {
    System.out.println("Insufficient arguments");
   }
finally
{
 res = a + b;
 System.out.println("Addition is "+res);
}
  }
}
```

Executes even if exception is not handled

Output



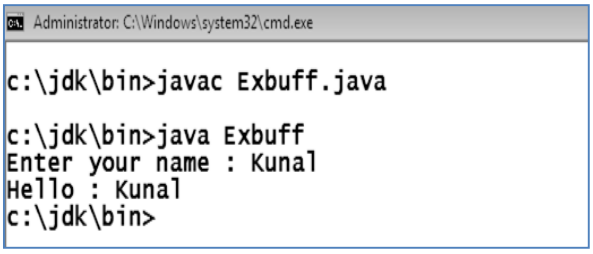
Program 12 : Write programs for implementation of throw,throws clause.

Soln. :

```
import java.io.*;
class Exbuff
{
public static void main(String ar[ ]) throws IOException
{
String nm;
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
System.out.print("Enter your name : ");
nm = br.readLine( );
System.out.print("Hello : "+nm);
}
}
```

Creating object for BufferedReader

Output



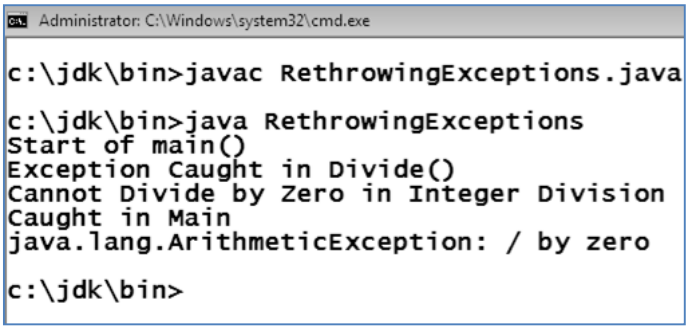
Programs for implementation of throw, throws clause. Part – II.

```
class test
{
    static void divide( )
  {
    int x,y,z;
    try
```

```
    {
      x = 5 ;
      y = 0 ;
      z = x/y ;
      System.out.println(x + "/"+ y +" = " + z);
    }
    catch(ArithmeticException e)
    {
     System.out.println("Exception Caught in Divide( )");
     System.out.println("Cannot Divide by Zero in Integer Division");
     throw e
    }
  }
 }
public class RethrowingExceptions
{
  public static void main(String ar[ ])
   {
       System.out.println("Start of main( )");
       try
       {
          test.divide( );
       }
     catch(ArithmeticException e)
       {
         System.out.println("Caught in Main");
         System.out.println(e);
       }
   }
}
```

Re-throwing the exception

Output



Program 13 : Write programs using multithreading.

Soln. :

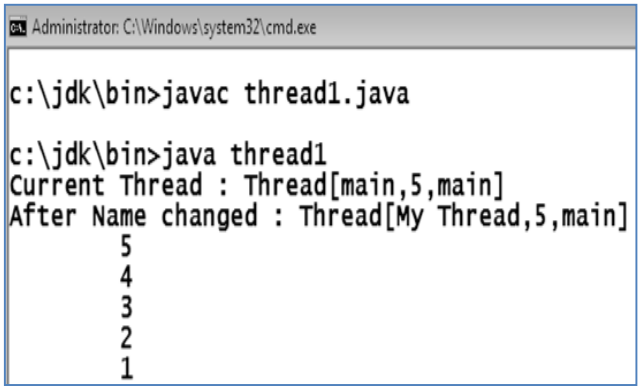
Program for implementation of multhreading operation Part - I.

```
class thread1
{
public static void main(String ar[ ])
{
Thread t = Thread.currentThread( );
System.out.println("Current Thread : "+t);
t.setName("My Thread");
System.out.println("After Name changed : "+t);
try
{
for(int n=5;n>0;n--)
{
System.out.println("\t"+n);
Thread.sleep(1000);
}
}
catch(InterruptedException e)
{
System.out.println("Thread interrupted");
}
}
}
```

t prints : name of thread(default-main), priority (default-5), group of thread(default-main)

Sleep( ) halts the execution for 1 second

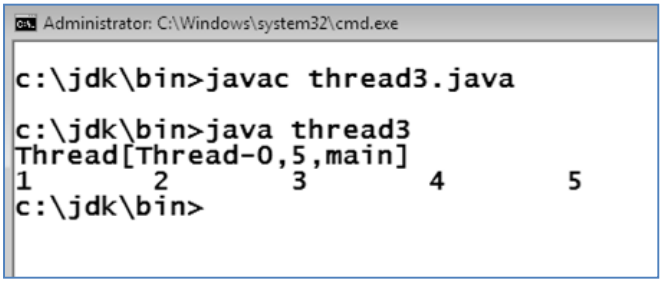
Output



Program for implementation of multithreading operation Part - II.

```
class MyThread extends Thread
{
public void run( )
{
for(int i=1;i<6;i++)
System.out.print(i+"\t");
}
}
class thread3
{
public static void main(String ar[ ])
{
MyThread t = new MyThread( );
System.out.println(t);
t.start( );
}
}
```

Output



Program 14 : Write program to design any type of form using AWT components.

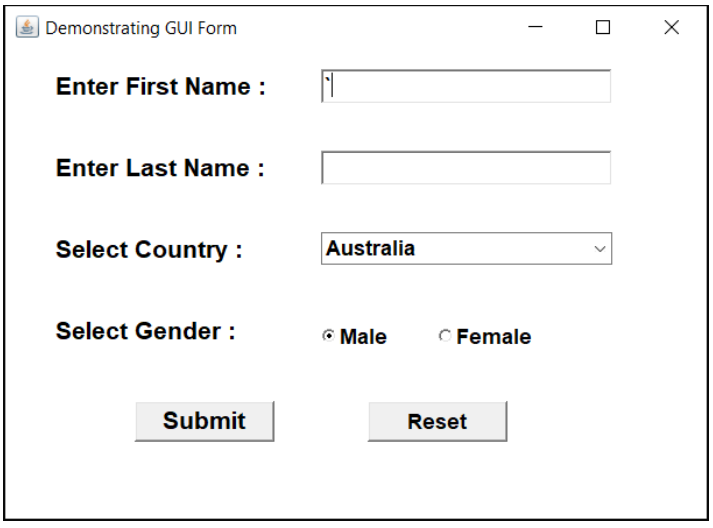
Soln. :

```
import java.awt.*;
class GUIForm extends Frame
{
    public GUIForm( )
    {
        setLayout(null);
        Font f1 = new Font("Arial", Font.BOLD, 21);
        Font f2 = new Font("Times New Roman", Font.BOLD, 18);
```

```
Label lb1 = new Label("Enter First Name : ");
lb1.setFont(f1);
Label lb2 = new Label("Enter Last Name : ");
lb2.setFont(f1);
Label lb3 = new Label("Select Country : ");
lb3.setFont(f1);
Label lb4 = new Label("Select Gender : ");
lb4.setFont(f1);

TextField tf1 = new TextField(15);
tf1.setFont(f2);
TextField tf2 = new TextField(15);
tf2.setFont(f2);

Choice ch = new Choice( );
ch.setFont(f2);
ch.add("Australia");
ch.add("Bangladesh");
ch.add("India");
ch.add("Nepal");
ch.add("SriLanka");

CheckboxGroup gr = new CheckboxGroup( );
Checkbox ch1 = new Checkbox("Male", true, gr);
ch1.setFont(f2);
Checkbox ch2 = new Checkbox("Female", false, gr);
ch2.setFont(f2);

Button btn1 = new Button("Submit");
btn1.setFont(f1);
Button btn2 = new Button("Reset");
btn2.setFont(f2);

add(lb1); add(lb2); add(lb3); add(lb4);
add(tf1); add(tf2); add(ch1); add(ch2); add(ch);
add(btn1); add(btn2);
```

```
        lb1.setBounds(50, 50, 190, 40);
        lb2.setBounds(50, 120, 190, 40);

        tf1.setBounds(280, 55, 250, 30);
        tf2.setBounds(280, 125, 250, 30);

        lb3.setBounds(50, 190, 190, 40);
        lb4.setBounds(50, 260, 190, 40);

        ch.setBounds(280, 195, 250, 40);
        ch1.setBounds(280, 265, 100, 40);
        ch2.setBounds(380, 265, 100, 40);

        btn1.setBounds(120, 340, 120, 35);
        btn2.setBounds(320, 340, 120, 35);
    }
    public static void main(String ar[ ])
    {
        GUIForm fr = new GUIForm( );
        fr.setSize(620,450);
        fr.setTitle("Demonstrating GUI Form");
        fr.setVisible(true);
    }
}
```

The output of above program will be :



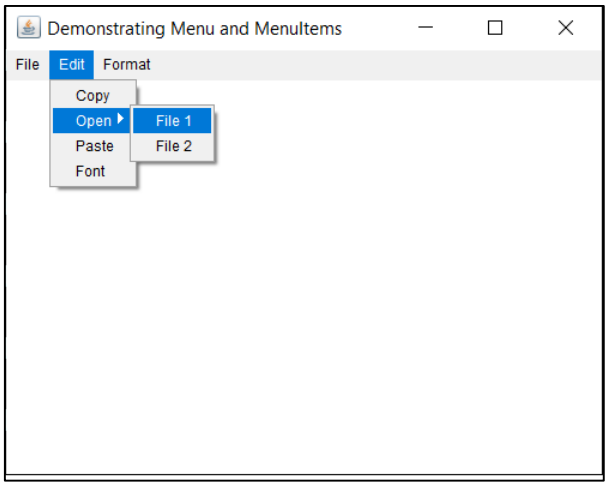
Program 15 : Write program to create a menu bar with variousmenu items and sub menu items.

Soln. :

```
import java.awt.*;
class MenuDemo extends Frame
{
public MenuDemo( )
{
MenuBar mb = new MenuBar( );
Menu m1 = new Menu("File");
Menu m2 = new Menu("Edit");
Menu m3 = new Menu("Format");
Menu m4 = new Menu("Open");
MenuItem mi1 = new MenuItem("New");
MenuItem mi2 = new MenuItem("Save");
MenuItem mi3 = new MenuItem("Save As");
MenuItem mi4 = new MenuItem("Cut");
MenuItem mi5 = new MenuItem("Copy");
MenuItem mi6 = new MenuItem("Paste");
MenuItem mi7 = new MenuItem("Font");
MenuItem mi8 = new MenuItem("File 1");
MenuItem mi9 = new MenuItem("File 2");
CheckboxMenuItem cmi1 = new CheckboxMenuItem("Word Wrap",true);
mb.add(m1);
mb.add(m2);
mb.add(m3);
m1.add(mi1);
m1.add(mi2);
m1.add(mi3);
m1.add(mi4);
m2.add(mi5);
m2.add(m4);        // nesting of menu
m2.add(mi6);
m2.add(mi7);
m3.add(cmi1);
m4.add(mi8);
m4.add(mi9);
setMenuBar(mb);
}
```

```
public static void main(String ar[ ])
{
MenuDemo fr = new MenuDemo( );
fr.setSize(400, 400);
fr.setTitle("Demonstrating Menu and MenuItems");
fr.setVisible(true);
}
}
```

The output of above program will be :



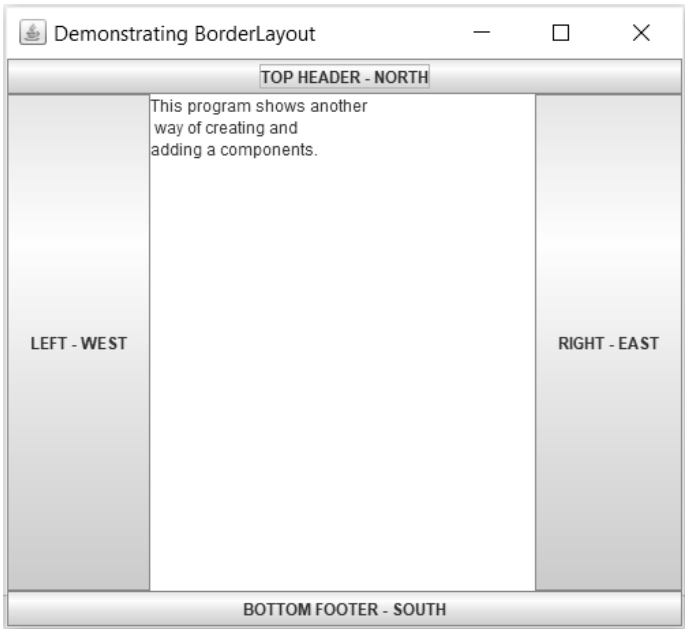
Program 16 : Write program to demonstrate the use of border layout. The layout shows four buttons at four sides with captions “left”, “right”, “top” and “bottom” using Swing Components.

Soln. :

```
import javax.swing.*;
import java.awt.BorderLayout;
import java.awt.Font;
public class BorderLayoutDemo extends JFrame
{
    public BorderLayoutDemo( )
    {
        setLayout(new BorderLayout( ));
        JButton bt1 = new JButton("TOP HEADER - NORTH");
        add(bt1, BorderLayout.NORTH);

        JButton btn2 = new JButton("BOTTOM FOOTER - SOUTH");
        add(btn2 ,BorderLayout.SOUTH);

        JButton btn3 = new JButton("RIGHT - EAST");
```

```
        add(btn3, BorderLayout.EAST);

        JButton btn4 = new JButton("LEFT - WEST");
        add(btn4, BorderLayout.WEST);

        String s = "This program shows another\n" + " way of creating and\n" + "adding components.\n";
        JTextArea ta = new JTextArea(s);
        add(ta, BorderLayout.CENTER);
    }
    public static void main(String ar[ ])
    {
        BorderLayoutDemo fr = new BorderLayoutDemo( );
        fr.setSize(400, 400);
        fr.setTitle("Demonstrating BorderLayout");
        fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        fr.setVisible(true);
    }
}
```

The output of above program will be :



Program 17 : Write program to design a calculator to demonstrate the use of grid layout using swingcomponents.

Soln. :

```
import java.awt.*;
import javax.swing.*;
public class GridLayoutDemo extends JFrame
```

```
{
public GridLayoutDemo( )
{
Container ct = getContentPane( );
GridLayout gl = new GridLayout(4,3);
ct.setLayout(gl);
for(int i = 1;i <=9; i++)
{
        ct.add(new JButton("" + i));
}
JButton bt1 = new JButton("*");
JButton bt2 = new JButton("0");
JButton bt3 = new JButton("#");
ct.add(bt1);
ct.add(bt2);
ct.add(bt3);
}
        public static void main(String ar[ ])
        {
                GridLayoutDemo fr = new GridLayoutDemo( );
                fr.setSize(400, 400);
                fr.setTitle("Demonstrating GridLayout");
                fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                fr.setVisible(true);
        }
}
```

The output of above program will be :

Program 18 : Write program using swing to display a JComboBoxin a JFrame .

Soln. :

```
import java.awt.*;
import javax.swing.*;

public class JComboBoxDemo extends JFrame
{
public JComboBoxDemo( )
{
Container ct = getContentPane( );
ct.setLayout(null);

// Creating combo box
JLabeljl = new JLabel("Select Country :");
JComboBoxjc = new JComboBox( );
jc.addItem("France");
jc.addItem("Germany");
jc.addItem("Italy");
jc.addItem("Japan");

//adding combo box to content pane
ct.add(jl);
ct.add(jc);

//aligning components
jl.setBounds(30,50,100,30);
jc.setBounds(150,50,100,30);
}
public static void main(String ar[ ])
{
JComboBoxDemo fr1 = new JComboBoxDemo( );
fr1.setTitle("Demonstrating combo box");
fr1.setSize(300,400);
fr1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
fr1.setVisible(true);
}
}
```

The output of above program will be :

Program 19 : Write program to create JTree and JTable

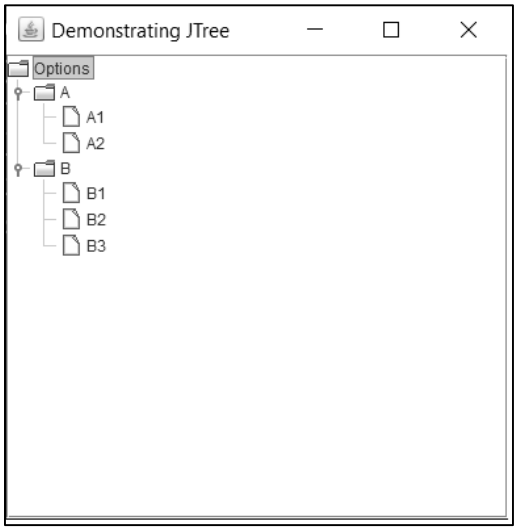
Soln. :

Program 19.1 : Program to create JTree.

```
import java.awt.*;
import javax.swing.*;
import javax.swing.tree.*;

public class JTreeDemo extends JFrame
{
public JTreeDemo( )
{
// Get content pane
Container ct = getContentPane( );

// Set layout manager
ct.setLayout(new BorderLayout( ));

// Create top node of tree
DefaultMutableTreeNode root = new DefaultMutableTreeNode("Options");

// Createsubtree of "A"
DefaultMutableTreeNode a = new DefaultMutableTreeNode("A");
root.add(a);
DefaultMutableTreeNode a1 = new DefaultMutableTreeNode("A1");
a.add(a1);
```

```
DefaultMutableTreeNode a2 = new DefaultMutableTreeNode("A2");
a.add(a2);

// Createsubtree of "B"
DefaultMutableTreeNode b = new DefaultMutableTreeNode("B");
root.add(b);
DefaultMutableTreeNode b1 = new DefaultMutableTreeNode("B1");
b.add(b1);
DefaultMutableTreeNode b2 = new DefaultMutableTreeNode("B2");
b.add(b2);
DefaultMutableTreeNode b3 = new DefaultMutableTreeNode("B3");
b.add(b3);

// Create tree
JTree tree = new JTree(root);

// Add tree to a scroll pane
int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
JScrollPane  jsp = new  JScrollPane(tree, v, h);

// Add scroll pane to the content pane
ct.add(jsp, BorderLayout.CENTER);
}

      public static void main(String ar[ ])
      {
            JTreeDemo fr = new JTreeDemo( );
            fr.setSize(400, 400);
            fr.setTitle("Demonstrating BorderLayout");
            fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            fr.setVisible(true);
      }

}
```

The output of above program will be :

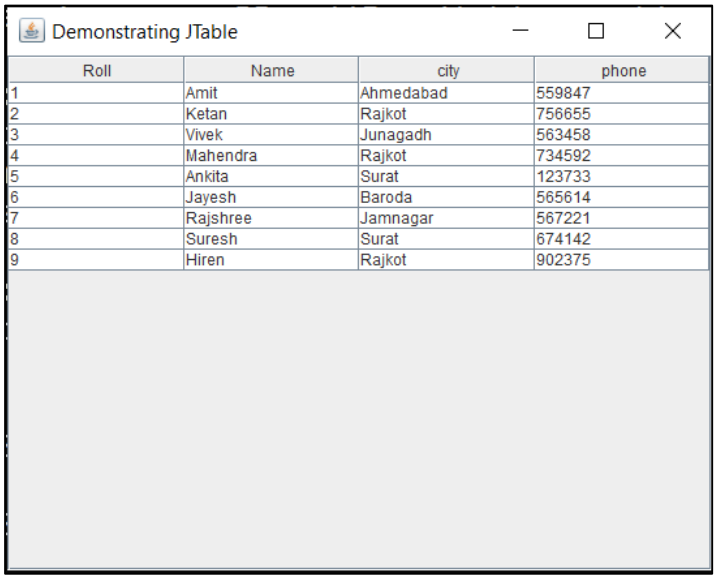


Program 19.2 : Program to create *JTable*.

```
importjava.awt.*;
importjavax.swing.*;

public class JTableDemo extends JFrame
{
public JTableDemo( )
{
// Get content pane
Container ct = getContentPane( );

// Set layout manager
ct.setLayout(new BorderLayout( ));

// Initialize column headings
final String[ ] colHeads = {"Roll", "Name", "city", "phone"};

// Initialize data
final Object[ ][ ] data = {
{ "1", "Amit", "Ahmedabad", "559847" },
{ "2", "Ketan", "Rajkot", "756655" },
{ "3", "Vivek", "Junagadh", "563458" },
{ "4", "Mahendra", "Rajkot", "734592" },
{ "5", "Ankita", "Surat", "123733" },
{ "6", "Jayesh", "Baroda", "565614" },
```

```
{ "7", "Rajshree", "Jamnagar", "567221" },
{ "8", "Suresh","Surat", "674142" },
{ "9", "Hiren", "Rajkot", "902375" }
};
// Create the table
JTable table = new JTable(data, colHeads);

// Add table to a scroll pane
int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
JScrollPanejsp = new JScrollPane(table, v, h);

// Add scroll pane to the content pane
ct.add(jsp, BorderLayout.CENTER);
}
        public static void main(String ar[ ])
        {
                JTableDemo fr = new JTableDemo( );
                fr.setSize(400, 400);
                fr.setTitle("Demonstrating JTable");
                fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                fr.setVisible(true);
        }
}
```

The output of above program will be :



Program 20 : Write program to handle key events and mouseevents.

Soln. :

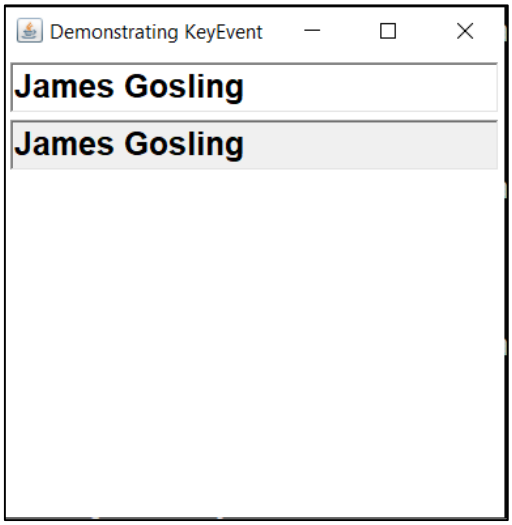
Program 20.1 : Demonstrating *KeyEvent*.

(Copy text of one TextField into another TextField, on keypress event)

```
import java.awt.*;
import java.awt.event.*;
class KeyEventDemo extends Frame implements KeyListener
{
	TextField tf1, tf2;

	public KeyEventDemo( )
	{
		setLayout(new FlowLayout( ));

		tf1 = new TextField(25);
		tf2 = new TextField(25);

		tf1.setFont( new Font("Arial", Font.BOLD, 25) );
		tf2.setFont( new Font("Arial", Font.BOLD, 25) );

		add(tf1); add(tf2);

		tf2.setEditable(false);
		tf1.addKeyListener(this);
	}

	public static void main(String ar[ ])
	{
		KeyEventDemo fr = new KeyEventDemo( );
		fr.setSize(400,400);
		fr.setTitle("Demonstrating KeyEvent");
		fr.setVisible(true);
	}

	public void keyReleased(KeyEvent ke)
	{
		String current_text = tf1.getText( );
```

```
            tf2.setText(current_text);
        }
        public void keyTyped(KeyEvent ke)
        { }
        public void keyPressed(KeyEvent ke)
        { }
}
```

The output of above program will be :



Program 20.2 : Demonstrating MouseEvent.

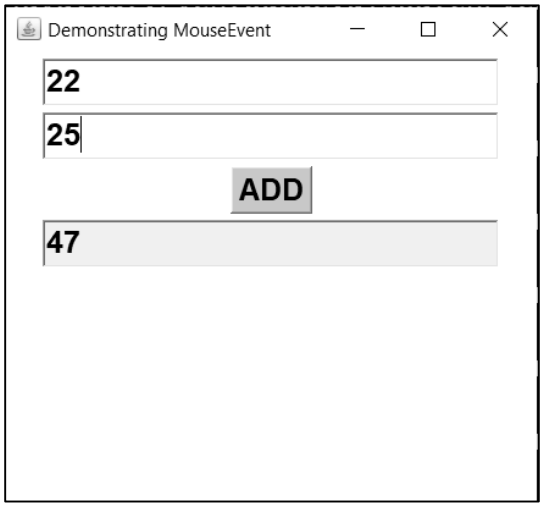
(Performing Addition when on mouse-enter event and clear TextFields on mouse-exit event)

```
import java.awt.*;
import java.awt.event.*;
class MouseEventDemo extends Frame implements MouseListener
{
        TextField tf1, tf2, tf3;
        Button btn;

        public MouseEventDemo( )
        {
            setLayout(new FlowLayout( ));
            Font f = new Font("Arial", Font.BOLD, 25);

            tf1 = new TextField(25);
            tf1.setFont(f);
            tf2 = new TextField(25);
```

```
        tf2.setFont(f);
        tf3 = new TextField(25);
        tf3.setFont(f);
        tf3.setEditable(false);
        btn = new Button("ADD");
        btn.setFont(f);
        btn.setBackground(Color.gray);

        add(tf1); add(tf2); add(btn); add(tf3);

        btn.addMouseListener(this);
    }
    public static void main(String ar[ ])
    {
        MouseEventDemo fr = new MouseEventDemo( );
        fr.setSize(400, 400);
        fr.setTitle("Demonstrating MouseEvent");
        fr.setVisible(true);
    }
    public void mouseEntered(MouseEvent me)
    {
        btn.setBackground(Color.cyan);
        int a = Integer.parseInt(tf1.getText( ) );
        int b = Integer.parseInt(tf2.getText( ) );

        int c = a + b;
        tf3.setText("" + c);
    }
    public void mouseExited(MouseEvent me)
    {
        btn.setBackground(Color.gray);
        tf1.setText(null);
        tf2.setText(null);
        tf3.setText(null);
    }
    public void mouseClicked(MouseEvent me)
    { }
```

```
    public void mousePressed(MouseEvent me)
    {}

    public void mouseReleased(MouseEvent me)
    {}
}
```

The output of above program will be :



Program 21 : Write program to implement action event in frameusing swing components.

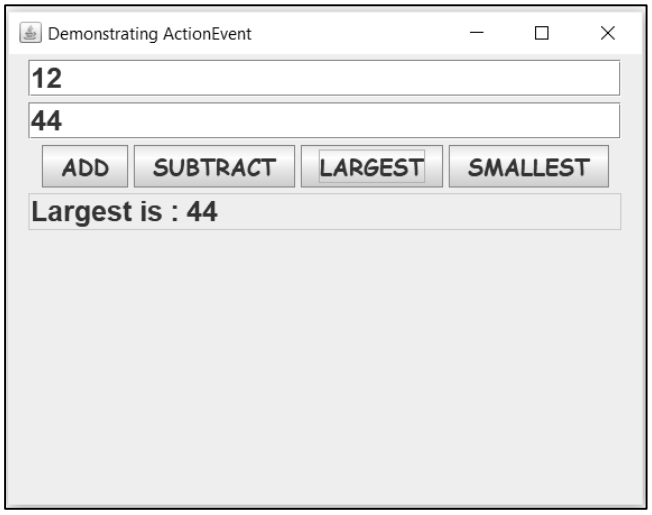
Soln. :

```
import java.awt.event.*;
import javax.swing.*;
import java.awt.*;
class ActionEventDemo extends JFrame
{
    JTextField tf1, tf2, tf3;
    JButton adbtn, subbtn, larbtn, smbtn;

    public ActionEventDemo( )
    {
        setLayout(new FlowLayout( ));
        Font f = new Font("Arial", Font.BOLD, 25);
        Font f2 = new Font("Comic Sans MS", Font.BOLD, 20);

        tf1 = new JTextField(25);
        tf1.setFont(f);
```

```
        tf2 = new JTextField(25);
        tf2.setFont(f);

        tf3 = new JTextField(25);
        tf3.setFont(f);
        tf3.setEditable(false);

        adbtn = new JButton("ADD");
        adbtn.setFont(f2);

        subbtn = new JButton("SUBTRACT");
        subbtn.setFont(f2);

        larbtn = new JButton("LARGEST");
        larbtn.setFont(f2);

        smbtn = new JButton("SMALLEST");
        smbtn.setFont(f2);

        add(tf1); add(tf2); add(adbtn); add(subbtn); add(larbtn); add(smbtn); add(tf3);

        adbtn.addActionListener(new Inner1( ) );
        subbtn.addActionListener(new Inner2( ) );
        larbtn.addActionListener(new Inner3( ) );
        smbtn.addActionListener(new Inner4( ) );
    }
    class Inner1 implements ActionListener
    {
        public void actionPerformed(ActionEvent ae)
        {
            int a = Integer.parseInt(tf1.getText( ) );
            int b = Integer.parseInt(tf2.getText( ) );
            int c = a + b;
            tf3.setText("Addition is : " + c);
        }
    }
```

```
    class Inner2 implements ActionListener
    {
        public void actionPerformed(ActionEvent ae)
        {
            int a = Integer.parseInt(tf1.getText( ) );
            int b = Integer.parseInt(tf2.getText( ) );
            int c = a - b;
            tf3.setText("Subtraction is : " + c);
        }
    }
    class Inner3 implements ActionListener
    {
        public void actionPerformed(ActionEvent ae)
        {
            int a = Integer.parseInt(tf1.getText( ) );
            int b = Integer.parseInt(tf2.getText( ) );
            int c = a > b ? a : b;
            tf3.setText("Largest is : " + c);
        }
    }
    class Inner4 implements ActionListener
    {
        public void actionPerformed(ActionEvent ae)
        {
            int a = Integer.parseInt(tf1.getText( ) );
            int b = Integer.parseInt(tf2.getText( ) );
            int c = a < b ? a : b;
            tf3.setText("Addition is : " + c);
        }
    }
    public static void main(String ar[ ])
    {
        ActionEventDemo fr = new ActionEventDemo( );
        fr.setSize(400, 400);
        fr.setTitle("Demonstrating ActionEvent");
        fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        fr.setVisible(true);
    }
}
```

The output of above program will be :



Program 22 : Write program to handle text event on swingcomponents.

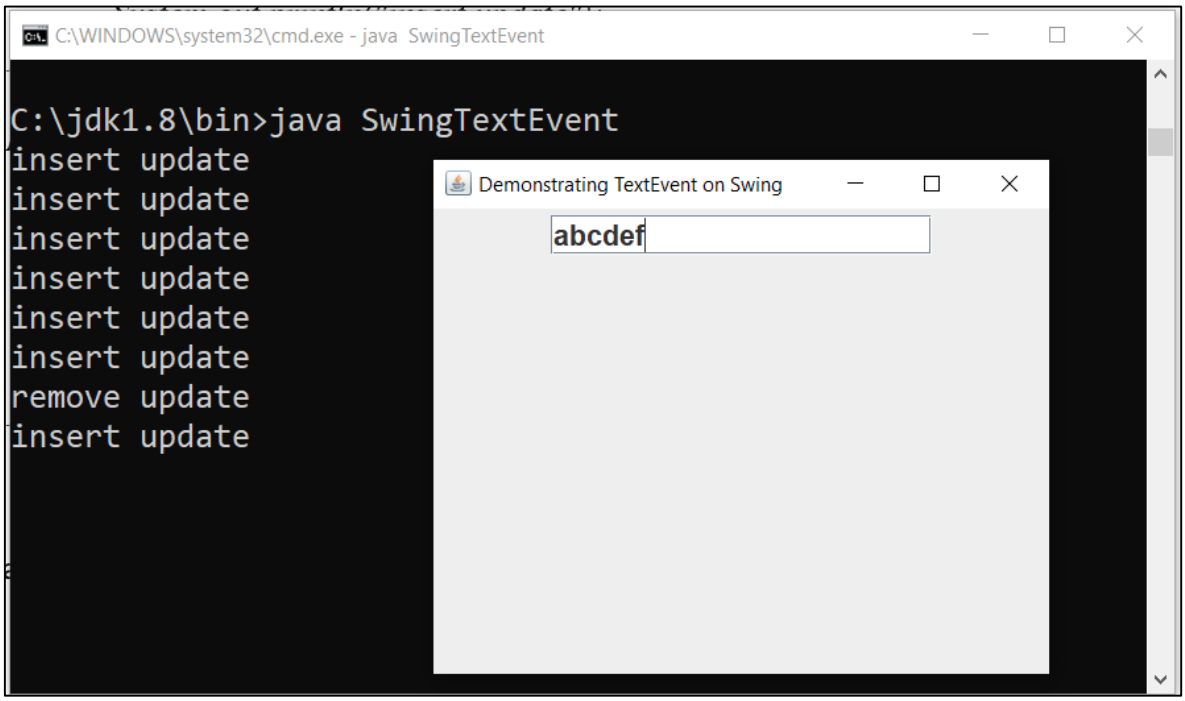
Soln. :

(TextEvent is not applicable on Swing's JTextField and JTextArea. Instead we have to use DocumentListener)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
class SwingTextEvent extends JFrame implements DocumentListener
{
	JTextField tf1;
	public SwingTextEvent( )
	{
		setLayout(new FlowLayout( ) );
		tf1 = new JTextField(15);
		tf1.setFont(new Font("Arial", Font.BOLD, 22) );

		add(tf1);

		tf1.getDocument( ).addDocumentListener(this);
	}

	public void changedUpdate(DocumentEvent e)
	{
		System.out.println("changed update");
	}
```

```
    public void removeUpdate(DocumentEvent e)
    {
        System.out.println("remove update");
    }

    public void insertUpdate(DocumentEvent e)
    {
        System.out.println("insert update");
    }

    public static void main(String ar[ ])
    {
        SwingTextEvent fr = new SwingTextEvent( );
        fr.setSize(400, 400);
        fr.setTitle("Demonstrating TextEvent on Swing");
        fr.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        fr.setVisible(true);
    }

}
```

The output of above program will be :

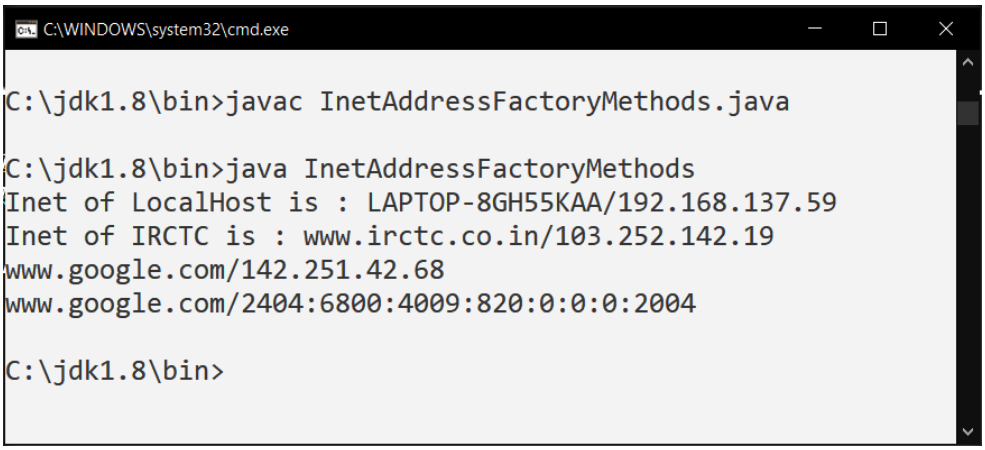


Program 23 : Write program to retrieve hostname and IP addressusing InetAddress class

Soln. :

```
import java.net.*;
class InetAddressFactoryMethods
{
	public static void main(String ar[ ]) throws UnknownHostException
	{
		InetAddress addr1 = InetAddress.getLocalHost( );
		System.out.println("Inet of LocalHost is : " + addr1);

		InetAddress addr2 = InetAddress.getByName("www.irctc.co.in");
		System.out.println("Inet of IRCTC is : " + addr2);

		InetAddress addr3[ ] = InetAddress.getAllByName("www.google.com");
		for (int i=0; i<addr3.length; i++)
		{
			System.out.println(addr3[i]);
		}
	}
}
```

The output of above program will be :



Program 24 : Write program to demonstrate various methods of:
URL class.
URLConnection.

Soln. :

Program 24.1 : Demonstrating methods of class *URL*.

```
import java.io.*;
import java.net.*;
```

```
class URLDemo
{
public static void main (String ar[ ]) throws IOException
{
URL url = new URL ("http://www.irctc.co.in");
System.out.println ("Authority = " + url.getAuthority( ));
System.out.println ("File = " +url.getFile( ));
System.out.println ("Host = " +url.getHost( ));
System.out.println ("Path = " +url.getPath( ));
System.out.println ("Port = " +url.getPort( ));
System.out.println ("Protocol = " +url.getProtocol( ));
System.out.println ("Query = " +url.getQuery( ));
System.out.println ("Ref = " +url.getRef( ));
System.out.println ("User Info = " +url.getUserInfo( ));
}
}
```

The output of above program will be :

Program 24.1 : Demonstrating methods of class *URLConnection*.

```
import java.io.*;
import java.net.*;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
public class URLConnectionDemo
{
public static void main(String ar[ ])
{
try
{
URL url = new URL("http://www.irctc.co.in");

//open the connection to the above URL.
URLConnection urlcon = url.openConnection( );

System.out.println("Result of getAllowUserInteraction( ) : " + urlcon.getAllowUserInteraction( ));
System.out.println("Result of getContentType( ) : " + urlcon.getContentType( ));
System.out.println("Result of getURL( ) : " + urlcon.getURL( ));
System.out.println("Result of getDoInput( ) : " + urlcon.getDoInput( ));
System.out.println("Result of getDoOutput( ) : " + urlcon.getDoOutput( ));
System.out.println("Result of getLastModified( ) : " + new Date(urlcon.getLastModified( )));

System.out.println("Result of getContentEncoding( ) : " + urlcon.getContentEncoding( ));

Map<String, List<String>> header = urlcon.getHeaderFields( );
for (Map.Entry<String, List<String>> mp : header.entrySet( ))
{
System.out.print(mp.getKey( ) + " : ");
System.out.println(mp.getValue( ).toString( ));
}
System.out.println( );

/*
Following code will print complete source code of ---> http://www.irctc.co.in

System.out.println("Complete source code of the URL is-");
System.out.println("---------------------------------");

BufferedReader br = new BufferedReader(new InputStreamReader(urlcon.getInputStream( )));
String i;
while ((i = br.readLine( )) != null)
```

```
{
System.out.println(i);
}
*/

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

The output of above program will be :

Program 25 : Write program that demonstrates connectionoriented communication using socket.

Soln. :

File 1 : ClientApplication.java

```
import java.io.*;
import java.net.*;
class ClientApplication
{
    public static void main(String ar[ ]) throws IOException, UnknownHostException
    {
        System.out.println("Client application Starts");
        InetAddress i = InetAddress.getLocalHost( );
        Socket client = new Socket(i, 100);
```

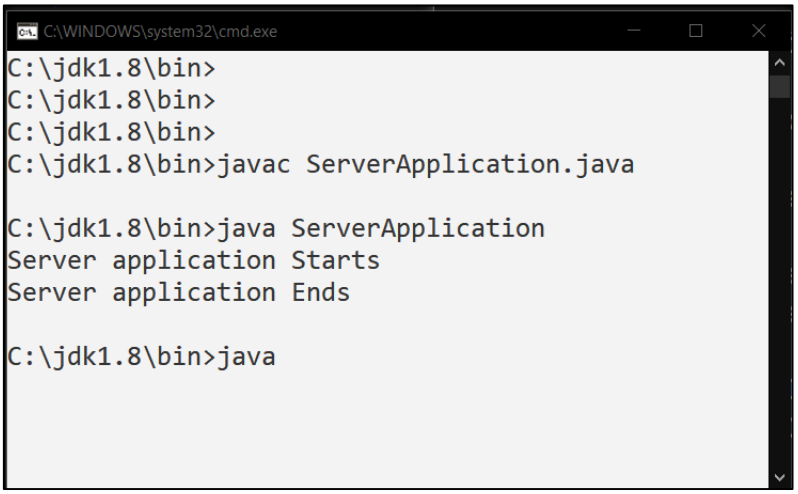
```
            DataInputStream dis = new DataInputStream(client.getInputStream( ));
            DataOutputStream dos = new DataOutputStream(client.getOutputStream( ));

            dos.writeUTF("5"); // sent 5 to obtain factorial of it
            // the request value is sent. Now client waits for response.

            String s2 = dis.readUTF( );
            System.out.println("Server Responded : " + s2);

            client.close( );

            System.out.println("Client application ends");
      }
}
```

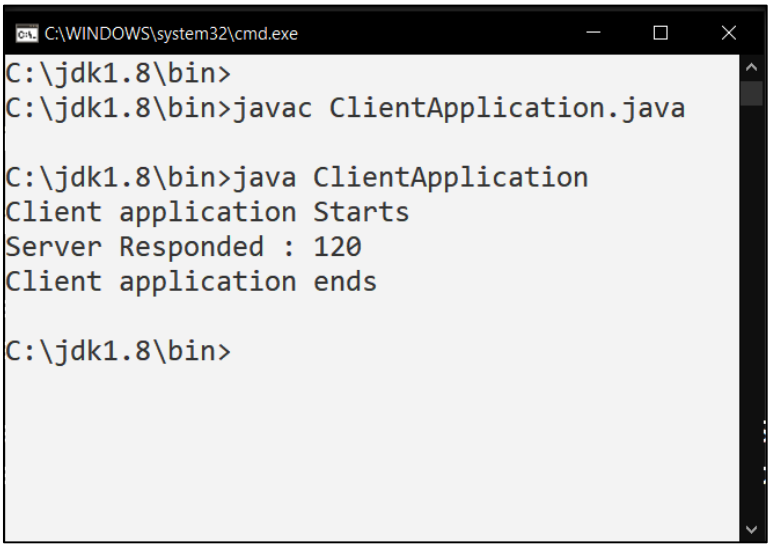
File 2 : ServerApplication.java

```
import java.net.*;
import java.io.*;
class ServerApplication
{
      public static void main(String ar[ ]) throws IOException
      {
            System.out.println("Server application Starts");

            ServerSocket ser = new ServerSocket(100);
            Socket soc = ser.accept( );

            DataInputStream dis = new DataInputStream(soc.getInputStream( ));
            DataOutputStream dos = new DataOutputStream(soc.getOutputStream( ));

            String s = dis.readUTF( );
            int num = Integer.parseInt(s);

            long f = 1;
            for(int i=1; i<=num; i++)
            {
                  f = f * i;
            }
```

```
            dos.writeUTF("" + f);


            soc.close( );
            ser.close( );


            System.out.println("Server application Ends");
        }
}
```

The output of above program will be : (First run Server application, then run Client application)





Program 26 : Write program to demonstrate sending and receiving data through datagram.

Soln. :

File 1 : DataSender.java

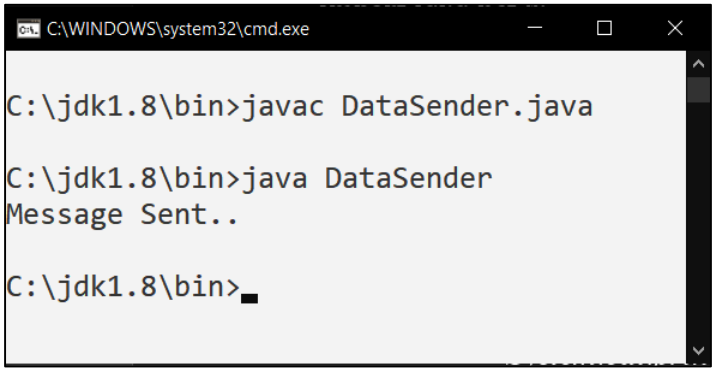
```
import java.net.*;
public class DataSender
```

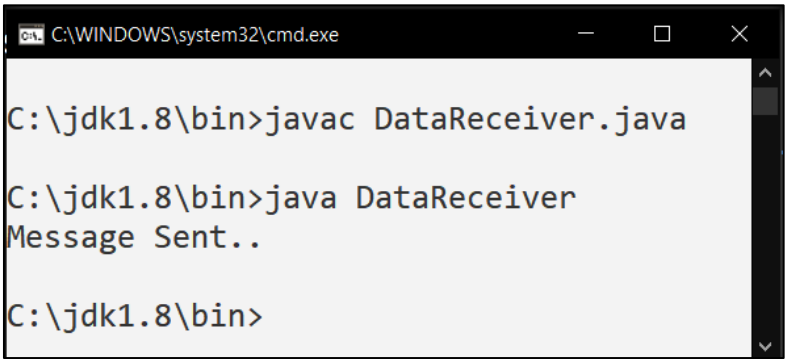
```
{
public static void main(String ar[ ]) throws Exception
{
DatagramSocket ds = new DatagramSocket( );
String str = "Welcome java";
InetAddress ip = InetAddress.getLocalHost( );
DatagramPacket dp = new DatagramPacket(str.getBytes( ), str.length( ), ip, 3000);
ds.send(dp);
ds.close( );
System.out.println("Message Sent..");
}
}
```

File 2 : DataReceiver.java

```
import java.net.*;
public class DataReceiver
{
public static void main(String ar[ ]) throws Exception
{
DatagramSocket ds = new DatagramSocket(3000);
byte buf[ ] = new byte[1024];
DatagramPacket dp = new DatagramPacket(buf, 1024);
ds.receive(dp);
String str = new String(dp.getData( ), 0, dp.getLength( ));
System.out.println(str);
ds.close( );
}
}
```

The output of above program will be : (Run Sender application first, then run receiver application)





Program 27 : Write program to : Create sample database.

Make connectivity with database

Soln. :

```
import java.sql.*;
class DatabaseDemo
{
	public static void main(String ar[ ])
	{
		try
		{
			// String query ="Write any insert / update / deleteQuery here";

			Class.forName("org.apache.derby.jdbc.ClientDriver");
			//loading JDBC driver for Derby Database

			Connection con = DriverManager.getConnection
("jdbc:derby://localhost:1527/sample", "app", "app");
			// obtaining connection with Database URL

			Statement st = con.createStatement( );
			// using interface Statement
			// st.executeUpdate(query);
			// uncomment above statement to fire query

			con.close( );
			//closing connection

			System.out.println("Connection Successful..!");
		}
```

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

Program 28 : Write program to implement following operationson database :

Insert record.

Update record.

Delete record.

Soln. :

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
class DatabaseOperations extends Frame
{
      TextField tf1, tf2, tf3, tf4;
      Label lb5;
        public DatabaseOperations( )
        {
              setLayout(null);
              Font f1 = new Font("Arial", Font.BOLD, 21);
              Font f2 = new Font("Times New Roman", Font.BOLD, 18);
        Font f3 = new Font("Times New Roman", Font.BOLD, 25);

              Label lb1 = new Label("Enter Employee ID : ");
              lb1.setFont(f1);
              Label lb2 = new Label("Enter Name : ");
              lb2.setFont(f1);
              Label lb3 = new Label("Enter Post : ");
              lb3.setFont(f1);
              Label lb4 = new Label("Enter Department : ");
              lb4.setFont(f1);
        lb5 = new Label("See Results Here");
                lb5.setFont(f3);
              tf1 = new TextField(15);
              tf1.setFont(f2);
```

```
        tf2 = new TextField(15);
        tf2.setFont(f2);
        tf3 = new TextField(15);
        tf3.setFont(f2);
        tf4 = new TextField(15);
        tf4.setFont(f2);

        Button btn1 = new Button("Insert Record");
        btn1.setFont(f1);
        Button btn2 = new Button("Update Record");
        btn2.setFont(f1);
        Button btn3 = new Button("Delete Record");
        btn3.setFont(f1);

        add(lb1); add(lb2); add(lb3); add(lb4);
        add(tf1); add(tf2); add(tf3); add(tf4);
        add(btn1); add(btn2); add(btn3); add(lb5);

        lb1.setBounds(90, 50, 190, 40);
        lb2.setBounds(90, 120, 190, 40);
        lb3.setBounds(90, 190, 190, 40);
        lb4.setBounds(90, 260, 190, 40);

        tf1.setBounds(320, 55, 280, 30);
        tf2.setBounds(320, 125, 280, 30);
        tf3.setBounds(320, 195, 280, 30);
        tf4.setBounds(320, 265, 280, 30);

        btn1.setBounds(60, 340, 180, 35);
        btn2.setBounds(290, 340, 180, 35);
        btn3.setBounds(500, 340, 180, 35);

    lb5.setBounds(200, 400, 250, 45);
    lb5.setForeground(Color.red);

        btn1.addActionListener(new Inner1( ) );
        btn2.addActionListener(new Inner2( ) );
        btn3.addActionListener(new Inner3( ) );
    }
```

```
    public static String fireQuery(String q, String m)
    {
      try
            {
                  String query = q;
                  Class.forName("org.apache.derby.jdbc.ClientDriver");
                  Connection con = DriverManager.getConnection
("jdbc:derby://localhost:1527/sample", "app", "app");
                  Statement st = con.createStatement( );
            st.executeUpdate(query);
      con.close( );
      return m;
            }
            catch (Exception e)
            {
      return e.getMessage( );
            }
    }
      class Inner1 implements ActionListener
      {
            public void actionPerformed(ActionEvent ae)
            {
            String s1 = tf1.getText( );
            String s2 = tf2.getText( );
            String s3 = tf3.getText( );
            String s4 = tf4.getText( );

            String q = "insert into emp_info values
(" + s1 + ",'" + s2 + "','" + s3 + "','" + s4 + "')";
            String res = fireQuery(q, "Employee Record Inserted..");
            lb5.setText(res);
            }
      }
      class Inner2 implements ActionListener
      {
            public void actionPerformed(ActionEvent ae)
            {
            String s1 = tf1.getText( );
            String s2 = tf2.getText( );
```

```
            String s3 = tf3.getText( );
            String s4 = tf4.getText( );
            String q = "update emp_info set
emp_name='" + s1 + "', emp_post='" +
s2 + "', emp_dept='" + s4 + "' where emp_id=" + s1;
            fireQuery(q, "Employee Record Updated..");
            }
      }
      class Inner3 implements ActionListener
      {
            public void actionPerformed(ActionEvent ae)
            {
            String s1 = tf1.getText( );
      String q = "delete from emp_info where emp_id=" + s1;
            fireQuery(q, "Employee Record Deleted");
            }
      }
      public static void main(String ar[ ])
      {
            DatabaseOperations fr = new DatabaseOperations( );
            fr.setSize(730, 500);
            fr.setTitle("Demonstrating Database Operations");
            fr.setVisible(true);
      }
}
```

The output of above program will be :

Program 29 : Write program to demonstrate the use of PreparedStatement.

Soln. :

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
class DatabaseOperations extends Frame
{
    TextField tf1, tf2, tf3, tf4;
    Label lb5;
      public DatabaseOperations( )
      {
            setLayout(null);
            Font f1 = new Font("Arial", Font.BOLD, 21);
            Font f2 = new Font("Times New Roman", Font.BOLD, 18);
     Font f3 = new Font("Times New Roman", Font.BOLD, 25);

            Label lb1 = new Label("Enter Employee ID : ");
            lb1.setFont(f1);
            Label lb2 = new Label("Enter Name : ");
            lb2.setFont(f1);
            Label lb3 = new Label("Enter Post : ");
            lb3.setFont(f1);
            Label lb4 = new Label("Enter Department : ");
            lb4.setFont(f1);
     lb5 = new Label("See Results Here");
     lb5.setFont(f3);

            tf1 = new TextField(15);
            tf1.setFont(f2);
            tf2 = new TextField(15);
            tf2.setFont(f2);
            tf3 = new TextField(15);
            tf3.setFont(f2);
            tf4 = new TextField(15);
            tf4.setFont(f2);

            Button btn1 = new Button("Insert Record");
            btn1.setFont(f1);
```

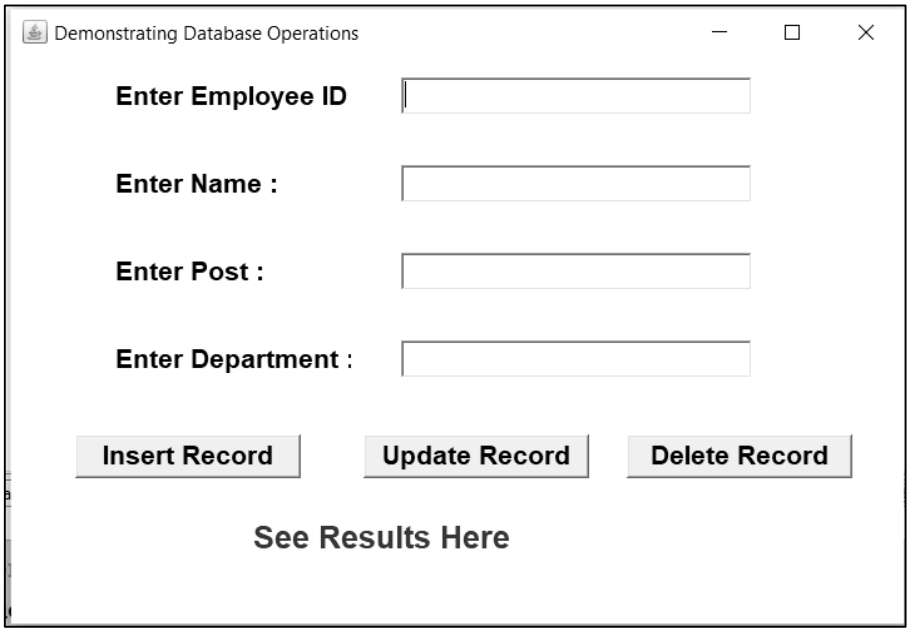
```
        Button btn2 = new Button("Update Record");
        btn2.setFont(f1);
        Button btn3 = new Button("Delete Record");
        btn3.setFont(f1);

        add(lb1); add(lb2); add(lb3); add(lb4);
        add(tf1); add(tf2); add(tf3); add(tf4);
        add(btn1); add(btn2); add(btn3); add(lb5);

        lb1.setBounds(90, 50, 190, 40);
        lb2.setBounds(90, 120, 190, 40);
        lb3.setBounds(90, 190, 190, 40);
        lb4.setBounds(90, 260, 190, 40);

        tf1.setBounds(320, 55, 280, 30);
        tf2.setBounds(320, 125, 280, 30);
        tf3.setBounds(320, 195, 280, 30);
        tf4.setBounds(320, 265, 280, 30);

        btn1.setBounds(60, 340, 180, 35);
        btn2.setBounds(290, 340, 180, 35);
        btn3.setBounds(500, 340, 180, 35);

    lb5.setBounds(200, 400, 250, 45);
    lb5.setForeground(Color.red);

        btn1.addActionListener(new Inner1( ) );
        btn2.addActionListener(new Inner2( ) );
        btn3.addActionListener(new Inner3( ) );
    }

    class Inner1 implements ActionListener
    {
        public void actionPerformed(ActionEvent ae)
        {
        int s1 = Integer.parseInt(tf1.getText( ));
    String s2 = tf2.getText( );
```

```
            String s3 = tf3.getText( );
            String s4 = tf4.getText( );
            try
            {
                        String query = "insert into emp_info values(?, ?, ?, ?)";
                        Class.forName("org.apache.derby.jdbc.ClientDriver");
                        Connection con = DriverManager.getConnection
("jdbc:derby://localhost:1527/sample", "app", "app");
                        PreparedStatement psmt = con.prepareStatement(query);

                psmt.setInt(1,s1);
                psmt.setString(2,s2);
                psmt.setString(3,s3);
                psmt.setString(4,s4);

                psmt.executeUpdate( );
                con.close( );
                lb5.setText("Employee Record Inserted..");
            }
            catch (Exception e)
            {
      lb5.setText(e.getMessage( ));
            }
            }
      }
      class Inner2 implements ActionListener
      {
            public void actionPerformed(ActionEvent ae)
            {
            int s1 = Integer.parseInt(tf1.getText( ));
            String s2 = tf2.getText( );
            String s3 = tf3.getText( );
            String s4 = tf4.getText( );

            try
            {
                  String query = "update emp_info set emp_name=?,
emp_post=?, emp_dept=? where emp_id=?";
```

```
                Class.forName("org.apache.derby.jdbc.ClientDriver");
                Connection con = DriverManager.getConnection
("jdbc:derby://localhost:1527/sample", "app", "app");
                PreparedStatement psmt = con.prepareStatement(query);

              psmt.setString(1,s2);
              psmt.setString(2,s3);
              psmt.setString(3,s4);
              psmt.setInt(4,s1);

              psmt.executeUpdate( );
              con.close( );
              lb5.setText("Employee Record Updated..");
      }
            catch (Exception e)
            {
      lb5.setText(e.getMessage( ));
            }
           }
      }
      class Inner3 implements ActionListener
      {
           public void actionPerformed(ActionEvent ae)
           {
            int s1 = Integer.parseInt(tf1.getText( ));
            try
            {
                String query = "delete from emp_info where emp_id=?";
                Class.forName("org.apache.derby.jdbc.ClientDriver");
                Connection con = DriverManager.getConnection
("jdbc:derby://localhost:1527/sample", "app", "app");
                PreparedStatement psmt = con.prepareStatement(query);

              psmt.setInt(1,s1);

              psmt.executeUpdate( );
              con.close( );
```

```
                lb5.setText("Employee Record Deleted..");
              }
              catch (Exception e)
              {
                lb5.setText(e.getMessage( ));
              }
             }
      }
      public static void main(String ar[ ])
      {
            DatabaseOperations fr = new DatabaseOperations( );
            fr.setSize(730, 500);
            fr.setTitle("Demonstrating Database Operations");
            fr.setVisible(true);
      }
}
```

The output of above program will be:



Program 30 : Write program to retrieve data from table using ResultSet interface.(Use various methods of navigation methods).

Soln. :

Program 30.1 : Program to demonstrate SELECT query to fetch every record from Employee Table and showing it in console.

```
import java.sql.*;
public class SelectQueryDemo1
```

```
{
   public static void main(String ar[])
   {
      String id, nm, pst, dep;
      try
      {
         String query = "select emp_id, emp_name, emp_post, emp_dept from emp_info";
         Class.forName("org.apache.derby.jdbc.ClientDriver");
         Connection con = DriverManager.getConnection
("jdbc:derby://localhost:1527/sample", "app", "app");
         Statement st = con.createStatement( );
         ResultSet rs = st.executeQuery(query);

         while(rs.next( ))
         {
            id = rs.getString("emp_id");
            nm = rs.getString("emp_name");
            pst = rs.getString("emp_post");
            dep = rs.getString("emp_dept");

            System.out.println(id + "\t" + nm + "\t" + pst + "\t" + dep);
         }
         con.close( );
      }
      catch(ClassNotFoundException cnfe)
      {
         System.out.println("Unable to Load JDBC Driver");
      }
      catch(SQLException sqe)
      {
         System.out.println(sqe.toString( ));
      }
   }
}
```

Program 30.2 : Demonstrating SELECT query to fetch record of specific employee and filling it in TextField objects.

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class SelectQueryDemo2 extends Frame implements ActionListener
{
  TextField tf1, tf2, tf3, tf4;
  Label lb5;
      public SelectQueryDemo2( )
      {
            setLayout(null);
            Font f1 = new Font("Arial", Font.BOLD, 21);
            Font f2 = new Font("Times New Roman", Font.BOLD, 18);
      Font f3 = new Font("Times New Roman", Font.BOLD, 25);

            Label lb1 = new Label("Enter Employee ID : ");
            lb1.setFont(f1);
            Label lb2 = new Label("Employee Name : ");
            lb2.setFont(f1);
            Label lb3 = new Label("Employee Post : ");
            lb3.setFont(f1);
            Label lb4 = new Label("Employee Department : ");
            lb4.setFont(f1);
      lb5 = new Label("See Exception Result Here.");
      lb5.setFont(f1);

            tf1 = new TextField(15);
            tf1.setFont(f2);
            tf2 = new TextField(15);
            tf2.setFont(f2);
      tf2.setEditable(false);
            tf3 = new TextField(15);
            tf3.setFont(f2);
      tf3.setEditable(false);
            tf4 = new TextField(15);
            tf4.setFont(f2);
```

```
        tf4.setEditable(false);

            Button btn1 = new Button("Fetch Record");
            btn1.setFont(f1);

            add(lb1); add(lb2); add(lb3); add(lb4);
            add(tf1); add(tf2); add(tf3); add(tf4);
            add(btn1); add(lb5);

            lb1.setBounds(90, 50, 190, 40);
            lb2.setBounds(90, 120, 190, 40);
            lb3.setBounds(90, 190, 190, 40);
            lb4.setBounds(90, 260, 190, 40);

            tf1.setBounds(320, 55, 280, 30);
            tf2.setBounds(320, 125, 280, 30);
            tf3.setBounds(320, 195, 280, 30);
            tf4.setBounds(320, 265, 280, 30);

        lb5.setBounds(200, 400, 250, 45);
        lb5.setForeground(Color.red);

        btn1.setBounds(230, 340, 180, 45);
            btn1.addActionListener(this);
        }
      public void actionPerformed(ActionEvent ae)
      {
         try
         {
            String query = "select emp_name, emp_post, emp_dept from
emp_info where emp_id=" + tf1.getText( );
            Class.forName("org.apache.derby.jdbc.ClientDriver");
            Connection con = DriverManager.getConnection
("jdbc:derby://localhost:1527/sample", "app", "app");
            Statement st = con.createStatement( );
            ResultSet rs = st.executeQuery(query);
            if(rs.next( ))
            {
               tf2.setText(rs.getString("emp_name"));
```

```
            tf3.setText(rs.getString("emp_post"));
            tf4.setText(rs.getString("emp_dept"));
          }
          else
          {
            lb5.setText("Invalid Employee ID");
          }
        }
        catch(ClassNotFoundException cnfe)
        {
          lb5.setText(cnfe.toString( ));
        }
        catch(SQLException sqe)
        {
          lb5.setText(sqe.toString( ));
        }
      }
      public static void main(String ar[ ])
       {
            SelectQueryDemo2 fr = new SelectQueryDemo2( );
fr.setSize(730, 500);
            fr.setTitle("Demonstrating use of Select Query");
            fr.setVisible(true);
       }
}
```

The output of above program will be :