Sem III Core Java Practical Doc 2020-2021

Practical No 1

Write a Java program to illustrate the concept of instance variable.

```
class DemoAreaVolume
{
    int l,m;
    double aS,aR;

    DemoAreaVolume(int a,int b,double c,double d)
    {
        l=a;
        m=b;
        aS=c;
        aR=d;
    }

    int areaS(int a)
    {
        return a*a;
    }

    int areaR(int a,int b)
    {
        return a*b;
    }

    public static void main(String args[])
    {
        DemoAreaVolume av = new DemoAreaVolume(10,10,0.0,0.0);
        av.aS = av.areaS(av.l);
        System.out.println("\nArea of square: "  +av.aS);
        av.aR = av.areaR(av.l,av.m);
        System.out.println("\nArea of rectangle: " +av.aR);

    }
}
```

Output:

Area of square: 100.0
Area of rectangle: 100.0

Practical No 2

Write a Java program to illustrate the concept of array.

Program 1.

```
class Array1D {
public static void main(String args[]) {
int month_days[];
month_days = new int[12];
month_days[0] = 31;
month_days[1] = 28;
month_days[2] = 31;
month_days[3] = 30;
month_days[4] = 31;

month_days[5] = 30;
month_days[6] = 31;

month_days[7] = 31;
month_days[8] = 30;
month_days[9] = 31;
month_days[10] = 30;
month_days[11] = 31;
System.out.println("April has " + month_days[3] + " days.");
}
}
```

Output:

April has 30 days.

Program 2.

```
// Demonstrate a two-dimensional array.
class Array2D {
public static void main(String args[]) {
int twoD[][]= new int[4][5];
int i, j, k = 0;
for(i=0; i<4; i++)
for(j=0; j<5; j++) {
twoD[i][j] = k;
k++;
}
for(i=0; i<4; i++) {
for(j=0; j<5; j++)
System.out.print(twoD[i][j] + " ");
System.out.println();
}
}
```

```
}
```

This program generates the following output:

```
0 1 2 3 4
5 6 7 8 9
10 11 12 13 14
15 16 17 18 19
```

Program 3:

```
// Demonstrate a three-dimensional array.
class Array3D {
public static void main(String args[]) {
int threeD[][][] = new int[3][4][5];
int i, j, k;
for(i=0; i<3; i++)
for(j=0; j<4; j++)
for(k=0; k<5; k++)
threeD[i][j][k] = i * j * k;
for(i=0; i<3; i++) {
for(j=0; j<4; j++) {
for(k=0; k<5; k++)
System.out.print(threeD[i][j][k] + " ");
System.out.println();
}
System.out.println();
}
}
}
```

This program generates the following output:

```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

0 0 0 0 0
0 1 2 3 4
0 2 4 6 8
0 3 6 9 12

0 0 0 0 0
0 2 4 6 8
0 4 8 12 16
0 6 12 18 24
```

Practical No 3

Write a Java program to illustrate the use of various string methods.

Program

```
//DemoString.java

class DemoString
{
  public static void main(String args[])
  {
  System.out.println("\n---------Demo of String class-------");

  char ch[]={'H','e','l','l','o',' ','T','h','e','r','e',' ','(','(','o','V','o',')',')'};
  String s1="Hello There (( o V o )) ";//First object
  String s2="Hello There (( o V o )) ";//Reference to first object
  String s3=new String("Hello There (( o V o )) ");//Second object
  String s4=new String(ch);//converting ch array to string

  System.out.println(s1);
  System.out.println(s2);
  System.out.println(s3);
  System.out.println(s4);

  System.out.println("\n--  Formated String ---");
  String name="Aarti";
  String sf1=String.format("name is %s",name);
  String sf2=String.format("value is %f",56.45675);
  String sf3=String.format("value is %20.12f",78.56768); //returns 12 char fractional part
filling with 0

  System.out.println(sf1);
  System.out.println(sf2);
  System.out.println(sf3);
```

```
  String str1 = String.format("%d", 101);     // Integer value
  String str2 = String.format("%s", "Amar Singh"); // String value
  String str3 = String.format("%f", 101.00);  // Float value
  String str4 = String.format("%x", 105);    // Hexadecimal value
  String str5 = String.format("%c", 'K');          // Char value

  System.out.println(str1);
  System.out.println(str2);
  System.out.println(str3);
  System.out.println(str4);
  System.out.println(str5);

//integer formating

System.out.println("\n -----integer formating-----");

String si1 = String.format("%d", 101);

// Specifying length of integer
String si2 = String.format("|%10d|", 101);

// Left-justifying within the specified width
String si3 = String.format("|%-10d|", 101);

String si4 = String.format("|% d|", 101);

// Filling with zeroes
String si5 = String.format("|%010d|", 101);

System.out.println(si1);
System.out.println(si2);
System.out.println(si3);
System.out.println(si4);
System.out.println(si5);

        System.out.println("------Substring Demo-----");
```

```
        String s11="Online Lectures :(  / :)";
        String substr1 = s11.substring(0); // Starts with 0 and goes to end
        System.out.println(substr1);
        String substr2 = s11.substring(16,18); // Starts from 16 and goes to 18
        System.out.println(substr2);
    //  String substr3 = s11.substring(16,30); // Returns Exception

System.out.println("string length is: "+substr1.length());
System.out.println("string contains :)  "+s11.contains(":)"));
System.out.println("Character at index 7 is : "+s11.charAt(7));

String se1="Java";
String se2="java";
String se3="Java";
System.out.println("----Case Sensetive----");
System.out.println(se1.equals(se2));
System.out.println(se1.equals(se3));

System.out.println("----Case InSensetive----");
System.out.println(se1.equalsIgnoreCase(se2));
System.out.println(se1.equalsIgnoreCase(se3));

System.out.println("----Concat----");
String fn="Aarti";
String ln="Pardeshi";
System.out.println(fn.concat(ln));

System.out.println("----Replace----");
System.out.println(fn.replace("A","Bh"));
System.out.println("Index of P in Last name is "+ln.indexOf("P"));
System.out.println("Surname in lower case : "+ln.toLowerCase());
System.out.println("Surname in upper case : "+ln.toUpperCase());
String strim1="     Are you feeling tired??      ";
System.out.println("Before Trim "+strim1);
```

```
System.out.println("After Trim "+strim1.trim());
 }//end main
 }//end class
```

Output:

```
C:\Program Files\Java\jdk1.7.0_51\bin>javac DemoString.java

C:\Program Files\Java\jdk1.7.0_51\bin>java DemoString

---------Demo of String class-------
Hello There (( o V o ))
Hello There (( o V o ))
Hello There (( o V o ))
Hello There ((oVo))

--  Formated String ---
name is Aarti
value is 56.456750
value is      78.567680000000
101
Amar Singh
101.000000
69
K

 -----integer formating-----
101
|       101|
|101       |
| 101|
|0000000101|
------Substring Demo-----
Online Lectures :(  / :)
:(
```

```
string length is: 24
string contains :)  true
Character at index 7 is : L
----Case Sensetive----
false
true
----Case InSensetive----
true
true
----Concat----
AartiPardeshi
----Replace----
Bharti
Index of P in Last name is 0
Surname in lower case : pardeshi
Surname in upper case : PARDESHI
Before Trim      Are you feeling tired??
After Trim Are you feeling tired??
```

Practical No 4

Write a Java program to illustrate the concept of package creation and its usage.

Write a Java program to create a package MyPack with the class Balance to check the account balance of user. If it is less than 0 then show message.

Step 1 :

Write a java program in the bin directory of jdk

```
//AccountBalance.java
package MyPack;

 class Balance{
  String name;
  double bal;

            Balance(String n,double b){
            name=n;
            bal=b;
           }

           void show(){
            if(bal<0)
            System.out.println("-->");
            System.out.println(name+" : Rs."+bal);
           }
}

  public class AccountBalance{
           public static void main(String args[]){
                    Balance  current[]=new Balance[3];
                    current[0]=new Balance("Aarti",123.23);
                    current[1]=new Balance("Shailesh",183.33);
                    current[2]=new Balance("Arun",-1.43);
                              for(int i=0;i<3;i++)
                                 current[i].show();
           }
}
```

Step 2 :
Compile the program using following command
C:\jdk1.3\bin>javac -d . AccountBalance.java

Step 3 :
Run the program using following command
C:\jdk1.3\bin>java MyPack.AccountBalance
Aarti : Rs.123.23

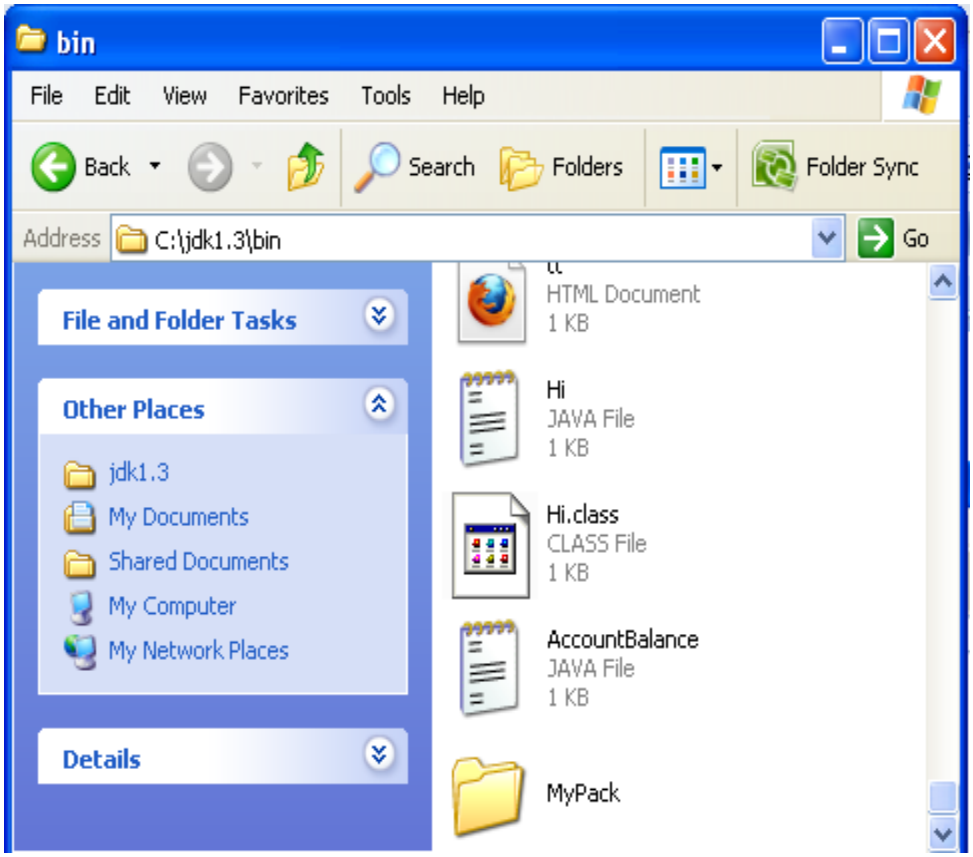
Shailesh : Rs.183.33
-->
Arun : Rs.-1.43

Explanation :

1. javac -d . Program_name.java

-d <directory> Specify where to place generated class files

The above command states that put Program_name.class in the current directory. Hence package MyPack will be automatically created with 2 class files AccountBalance.class and Balance.class. MyPack will be placed in jdk1.6\bin directory.



Write a java program to create a package and display a message.

```
//DemoPack.java

package SecondPack;


 class DemoPack

{

  public static void main(String args[])

 {
```

```
  System.out.println("This is SecondPack package");
 }
}
```

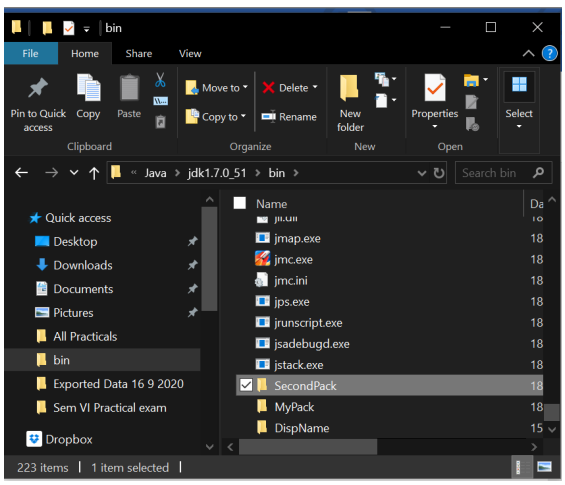
Output:

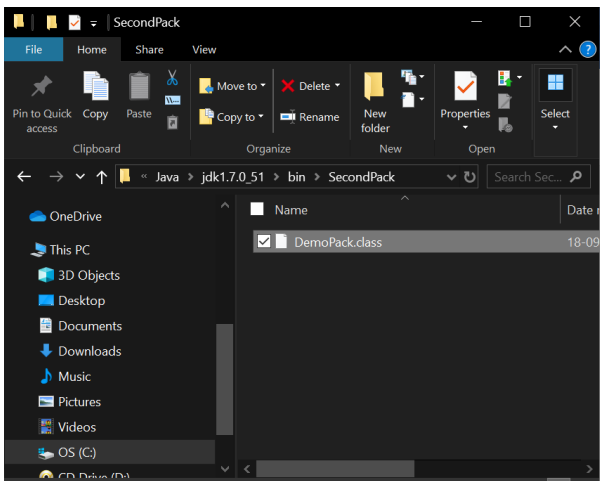
C:\Program Files\Java\jdk1.7.0_51\bin>javac -d . DemoPack.java

C:\Program Files\Java\jdk1.7.0_51\bin>java SecondPack.DemoPack

This is SecondPack package

C:\Program Files\Java\jdk1.7.0_51\bin>

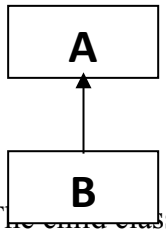




Practical No 5
Demonstrate Java inheritance using extends keyword.

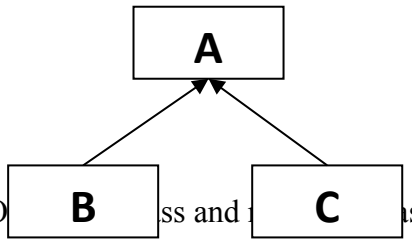
The one class can inherit or reuse the data members or member methods of another class. This technique is known as inheritance. The four types of inheritance are as follows :

1. Single Inheritance :



The child class has only one parent class or super class or base class.

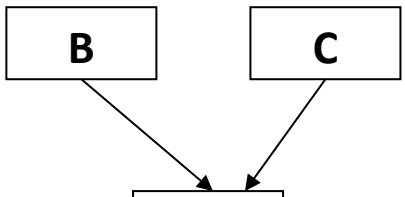
2. Hierarchical Inheritance



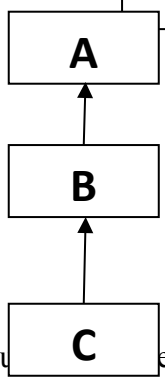
One class and two classes

3. Multiple Inheritance (supported only for interface)

Two or many super classes and only one child class



4. Multilevel Inheritance



Multiple inheritance can be obtained by connecting many single level inheritance. Multiple inheritance is supported in Java for only interface. Java does not support multiple inheritance for classes.

class Area

{

```
int width, height;
 Area(int w, int h)
 {
 width=w;
 height=h;
 }
 public void showArea()
 {
 int A;
 System.out.println("Height : "+height+" and width is "+width);
 A=width*height;
 System.out.println("\nArea is "+A);
 }
}
class Rectangle extends Area
{
 Rectangle(int w,int h)
 {
  super(w,h);
 }
  public static void main(String args[])
 {
 Rectangle R=new Rectangle(10,20);
 R.showArea();
 }
}
```

Output :

```
C:\Program Files\Java\jdk1.6.0_45\bin>javac Rectangle.java
C:\Program Files\Java\jdk1.6.0_45\bin>java Rectangle
Height : 20 and width is 10
```

Area is 200

Note :

super keyword has two forms. The first is to use to call super class constructor and the second is to used when subclass data member has same name as that of the super class; then the subclass member hide the super class member. To indicate them use super keyword.

Practical No 6
Demonstrate method overloading and method overriding in Java.

Method Overriding

When subclass defines a method having the same name, return type and same parameter list as that of super class then subclass method override the method of superclass.

Comparision

Overloading	Overriding
• Same method name	Same method name
• Different prototype	Same prototype
• Same class	Different class

Ex.

```
class DemoOverride
{
 int i;
  DemoOverride()
  {
  i=10;
  }

  public void display()
  {
  System.out.println("i "+i);
  }
}

 class DemoChildOverride extends DemoOverride
{
 int j;
  DemoChildOverride()
  {
  j=20;
  }
```

```
  public void display()
  {
  System.out.println("j : "+j);
  }
}
 class MOverride
{
  public static void main(String args[])
  {
  DemoChildOverride dco = new DemoChildOverride();
  dco.i=40;
  dco.j=60;
  dco.display();
  }
}
```

Output :

```
C:\Program Files\Java\jdk1.6.0_45\bin>javac MOverride.java
C:\Program Files\Java\jdk1.6.0_45\bin>java MOverride
j : 60
```

In above example there are two classes holding a parent child relation namely DemoOverride and DemoChildOverride respectively. The display method is present in both class with the same prototype. This situation illustrates the concept of method overriding hence whenever the call to the method will be made by child. The child's display method will be called.

Practical No 7
Demonstrate creating your own exception in Java.

Creating Your own exception subclasses

Java developer can define own exception class to handle specific/customized situation.

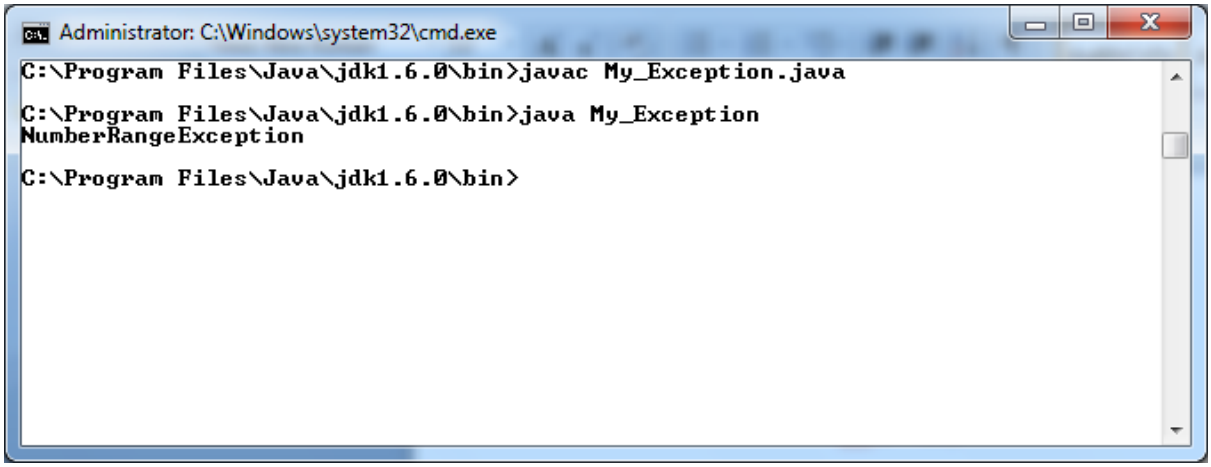
This is quite easy to do: just define a subclass of Exception. Subclasses don't need to actually implement anything—it is their existence in the type system that allows developer to use them as exceptions. The Exception class does not define any methods of its own. It does, of course, inherit those methods provided by Throwable. Thus, all exceptions, including those that developer create, have the methods defined by Throwable available to them.

In the following example we are creating a subclass NumberRangeException of class Exception. It accepts integer between 20 and 100. Others are rejected by throwing an exception.

//MyException.java

```
class NumberRangeException extends Exception
{
String msg;

NumberRangeException()
{
msg = new String("Enter a number between 20 and 100");
}
}

public class My_Exception
{
public static void main (String args [ ])
{
try
{
int x = 10;

if (x < 20 || x >100) throw new NumberRangeException( );
}
catch (NumberRangeException e)
{
System.out.println (e);
}
}
}
```

Output:



//Example No. 2

```
class MyException extends Exception
{
   MyException(String message)
  {
   super(message);
   }
}

public class ExceptionDemo
{
  public static void main(String args[]) throws Exception
 {
    ExceptionDemo exceptionDemo = new ExceptionDemo();
    exceptionDemo.displayNumbers();
  }

   public void displayNumbers() throws MyException
  {
   for(int i=0;i<10;i++)
   {
    System.out.println("i= "+i);
     if(i==6)
    {
     throw new MyException("My Exception Occurred");
     }
   }
 }
}
```

Output:

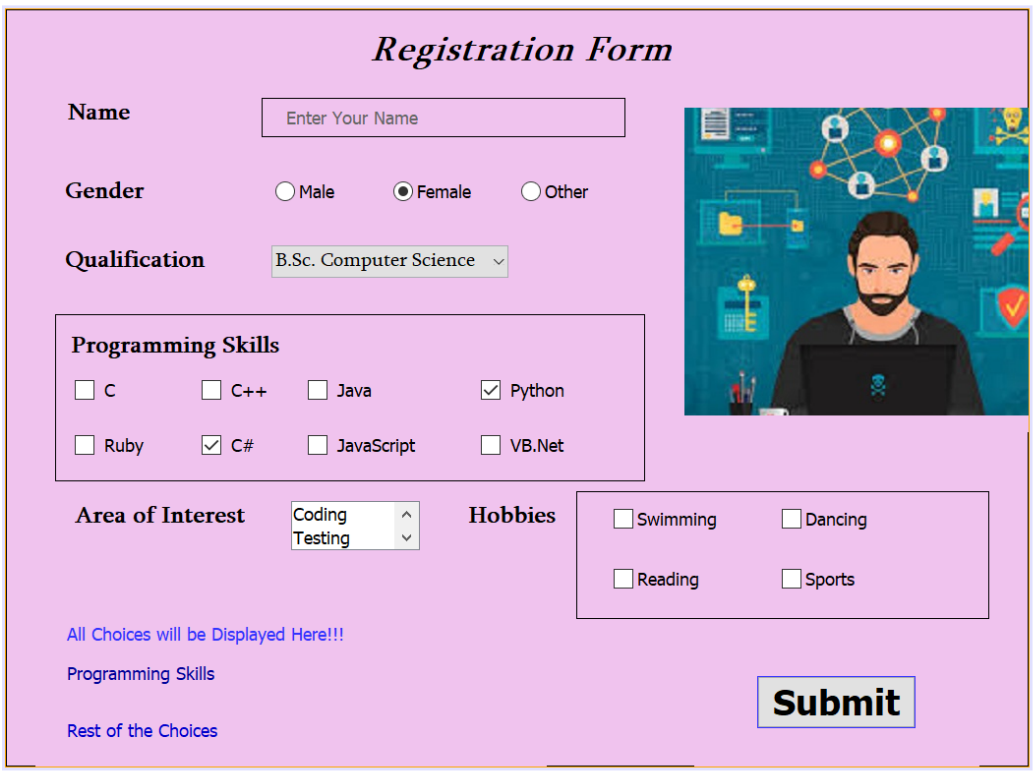
Select Administrator: C:\Windows\system32\cmd.exe

```
C:\Program Files\Java\jdk1.6.0\bin>javac ExceptionDemo.java

C:\Program Files\Java\jdk1.6.0\bin>java ExceptionDemo
i= 0
i= 1
i= 2
i= 3
i= 4
i= 5
i= 6
Exception in thread "main" MyException: My Exception Occurred
        at ExceptionDemo.displayNumbers(ExceptionDemo.java:24)
        at ExceptionDemo.main(ExceptionDemo.java:14)

C:\Program Files\Java\jdk1.6.0\bin>_
```

Practical No 8
Using various swing components design Java application to accept a student's resume. (Design form)

```
//Code On Submit
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if((jButton1.getActionCommand()).equals("Submit"))
        {
         String nm=jTextField1.getText();
         String gender="";
         if(jRadioButton1.isSelected())
             gender="Male";
         else if(jRadioButton2.isSelected())
             gender="Female";
         else gender="Other";

       //  String jcm1=jComboBox1.getActionCommand();
         String qf="",ps="";

        if(jComboBox1.getSelectedItem().equals("B.Sc. Computer Science"))
            qf="B.Sc. Computer Science";
        else if(jComboBox1.getSelectedItem().equals("B.Sc."))
            qf+="    B.Sc.";
        else if(jComboBox1.getSelectedItem().equals("B.A."))
            qf+="   B.A.";
        else if(jComboBox1.getSelectedItem().equals("B.Com"))
            qf+="B.Com";
        else
            qf="No qualification";

        if(jCheckBox1.isSelected())
            ps+=" C";
        if(jCheckBox2.isSelected())
            ps+=" C++";
        if(jCheckBox3.isSelected())
            ps+=" Java";
        if(jCheckBox4.isSelected())
            ps+=" Python";
        if(jCheckBox5.isSelected())
            ps+=" Ruby";
        if(jCheckBox6.isSelected())
            ps+=" C#";
        if(jCheckBox7.isSelected())
            ps+=" JavaScript";
        if(jCheckBox8.isSelected())
            ps+=" VB.Net";

        String ai="";
```

```
    if((jList1.getSelectedValue()).equalsIgnoreCase("Coding"))
        ai="Coding";
    else if((jList1.getSelectedValue()).equalsIgnoreCase("Testing"))
        ai="Testing";
    else if((jList1.getSelectedValue()).equalsIgnoreCase("Analysing"))
        ai="Analysing";
    else if((jList1.getSelectedValue()).equalsIgnoreCase("Management"))
        ai="Management";

    String hb="";
    if(jCheckBox12.isSelected())
        hb+=" Swimming";
    if(jCheckBox10.isSelected())
        hb+=" Dancing";
    if(jCheckBox9.isSelected())
        hb+=" Reading";
    if(jCheckBox11.isSelected())
        hb+=" Sports";

    jLabel8.setText(nm+" "+gender+" "+qf+" ");
    jLabel11.setText(" "+ps+" ");
    jLabel10.setText(" "+ai+" "+hb+" ");

    }
}
```



Practical No 9
Write a Java List example and demonstrate methods of Java List interface.

Practical No 10

Design simple calculator GUI application using AWT components.