**//program multiple inheritance with interface**

**//program for interface**

**//program from this and supper keyword**

**// define student class and data members name,roll,marks and display //percentage.**

import java.io.\*;

import java.lang.\*;

class MultipleInheritance

{

//instance variables declare

String name;

int roll;

double m1,m2;

//parameterized constructor

MultipleInheritance(String name,int roll,double m1,double m2)

{

this.name=name;

this.roll=roll;

this.m1=m1;

this.m2=m2;

}

public static void main(String []args)

{

//create object of child class i.e. student

student s=new student("abc",1,50,60);

s.calculate\_per();

s.display();

}

}

//interface-it is an class which contains abstract (incomplete)methods

interface exam

{

public void calculate\_per();

public void display();

}

//child class-inheritance concept

//extends+implements->multiple inheritance

class student extends MultipleInheritance implements exam

{

double result;

student(String name,int roll,double m1,double m2)

{

//cal to parent class constructor

super(name,roll,m1,m2);

}

public void calculate\_per()

{

result=((m1+m2)\*100)/200;

}

public void display()

{

System.out.println("Name : "+name);

System.out.println("Percentage is : "+result);

}

}

/\* output :

g:\Java>javac MultipleInheritance.java

g:\Java>java MultipleInheritance

Name : abc

Percentage is : 55.0

\*/

**//Q.List of Draw methods and set methods in applet :**

import java.applet.Applet;

import java.awt.\*;

/\* <applet code="GraphicsDemoDraw" width="300" height="300"> </applet> \*/

public class GraphicsDemoDraw extends Applet

{

public void paint(Graphics g)

{

//1.setFont(Font font): is used to set the graphics font to text

g.setFont(new Font("Times New Roman",Font.BOLD | Font.ITALIC,20));

//2.drawString(String str, int x, int y) :used to draw a text string

g.drawString("Welcome",40, 40);

//3.drawLine(int x1, int y1, int x2, int y2):

//is used to draw line between the points(x1, y1) and (x2, y2).

g.drawLine(20,30,20,250);

//4.drawRect(int x, int y, int width, int height):

//draws a rectangle with the points(x,y) & specified width and height.

g.drawRect(70,100,30,30);

//5.drawArc(int x, int y, int width, int height, int startAngle, int arcAngle):

//is used draw a circular or elliptical arc.

g.drawArc(90,150,30,30,30,270);

//6.drawOval(int x, int y, int width, int height):

//is used to draw oval with the specified width and height.

g.drawOval(70,200,50,50);

//7. drawPolygon(int x[],int y[],int length) : is used to draw a hollow polygon

int x[]={90,150,180,200,300};

int y[]={70,60,100,180,150};

int length=x.length;

g.drawPolygon(x,y,length);

//8.drawRoundRect(int x,int y,int width,int height,int WidthOfCornerAngle, int HeightOfCornerAngle) :

//is used to draw a hollow rectangle with rounded corners

g.drawRoundRect(150,250,30,30,10,10);

}

}

/\* To run applet program,use following commands:

g:\Java>javac GraphicsDemo.java

g:\Java>appletviewer GraphicsDemo.java

\*/

**//Q.List of fill methods of applet and set method :**

import java.applet.Applet;

import java.awt.\*;

/\* <applet code="GraphicsDemoFill" width="300" height="300"> </applet> \*/

public class GraphicsDemoFill extends Applet

{

public void paint(Graphics g)

{

//9.setColor(Color c): is used to set the graphics current color to the specified //color.

g.setColor(Color.pink);

//10.fillRect(int x, int y, int width, int height):

//is used to fill rectangle with the default color and specified width and height.

g.fillRect(170,100,50,50);

//11.fillOval(int x, int y, int width, int height):

//is used to fill oval with the default color and specified width and height.

g.fillOval(170,200,80,50);

//12.fillArc(int x, int y, int width, int height, int startAngle, int arcAngle):

//is used to fill a circular or elliptical arc.

g.fillArc(270,150,180,180,0,180);

//13.fillPolygon(int x[],int y[],int length)

//used to fillcolor in polygon, polygon may have any numbber number of sides,

//x[] indicates array of x-coordinate and y[] indicates array of y-coordinate

int x[]={20,50,100};

int y[]={50,150,200};

int n=x.length;

g.fillPolygon(x,y,n);

//14.fillRoundRect(int x,int y,int width,int height,int WidthOfCornerAngle, int HeightOfCornerAngle) :

//is used to fill color in round rectangle

g.fillRoundRect(170,100,50,50);

}

}

**//write a simple program which display message "welcome to java"**

import java.applet.\*;

import java.awt.\*;

/\*<applet code="WelcomeApplet" width="500" height="300"> </applet>\*/

public class WelcomeApplet extends Applet

{

public void paint(Graphics g)

{

g.setFont(new Font("Arial",Font.BOLD | Font.ITALIC,50));

g.drawString("Welcome to java",50,50);

}

}

**//Write a program to design an applet to display three circles filled with three different colors on screen**

import java.lang.\*;

import java.awt.\*;

import java.applet.Applet;

/\* <applet code="ThreeCircles" width="300" height="300"> </applet> \*/

public class ThreeCircles extends Applet

{

public void paint(Graphics g)

{

g.setColor(Color.red);

g.fillOval(20,70,100,100);

g.setColor(Color.green);

g.fillOval(120,70,100,100);

g.setColor(Color.blue);

g.fillOval(220,70,100,100);

}

}

**//Write a java program to display triangle filled with red color**

**//Polygon program**

import java.applet.\*;

import java.awt.\*;

/\*<applet code="TriangleApplet" width="300" height="300"> </applet> \*/

public class TriangleApplet extends Applet

{

public void paint(Graphics g)

{

int x[]={20,50,100};

int y[]={50,150,200};

int n=x.length;

g.setColor(Color.red);

g.fillPolygon(x,y,n);

//g.drawPolygon(x,y,n);

}

}

**//Write an applet to accept user name in the form of parameters and print 'Hello<username>'.**

**//Applet tag- parameter**

import java.applet.\*;

import java.awt.\*;

/\*<applet code="ParameterApplet" width=300 height=300>

<PARAM NAME="username" VALUE="abc">

</applet> \*/

public class ParameterApplet extends Applet

{

public void paint(Graphics g)

{

String str=getParameter("username");

g.drawString("Hello"+str,40,40);

}

}

**//Write an applet program that accept two input,strings using <Param> tag and concatenate the strings and display it in status window.**

import java.applet.\*;

import java.awt.\*;

/\*<applet code="ConcatApplet" width=300 height=300>

<PARAM NAME="name1" VALUE="Pallavi">

<PARAM NAME="name2" VALUE="Mhamane">

</applet> \*/

public class ConcatApplet extends Applet

{

public void paint(Graphics g)

{

String str=getParameter("name1");

String str1=getParameter("name2");

g.drawString(str.concat(str1),40,40);

}

}

**//Design an applet which accepts username as a parameter for html page and**

**// display** **number of character from it**

import java.applet.\*;

import java.awt.\*;

/\*<Applet code="NumOfCharApplet" height="300" width="300">

<PARAM NAME="username" value="abclldfghjk">

</Applet>\*/

public class NumOfCharApplet extends Applet

{

public void paint(Graphics g)

{

String xyz=getParameter("username");

String l=Integer.toString(xyz.length());

g.drawString(l,40,50);

}

}

**//Write a program to draw a bar chart for plotting students passing percentage in last 5 years**

import java.applet.\*;

import java.awt.\*;

/\*<Applet code="StudentBarChartApplet" height="300" width="300">

<PARAM NAME="m1" value="15">

<PARAM NAME="m2" value="25">

<PARAM NAME="m3" value="35">

<PARAM NAME="m4" value="45">

<PARAM NAME="m5" value="55">

<PARAM NAME="y1" value="2001">

<PARAM NAME="y2" value="2002">

<PARAM NAME="y3" value="2003">

<PARAM NAME="y4" value="2004">

<PARAM NAME="y5" value="2005">

</Applet>\*/

public class StudentBarChartApplet extends Applet{

public void paint(Graphics g)

{

int marks[]=new int[5];

String years[]=new String[5];

//convert string to integer

marks[0]=Integer.parseInt(getParameter("m1"));

marks[1]=Integer.parseInt(getParameter("m2"));

marks[2]=Integer.parseInt(getParameter("m3"));

marks[3]=Integer.parseInt(getParameter("m4"));

marks[4]=Integer.parseInt(getParameter("m5"));

years[0]=getParameter("y1");

years[1]=getParameter("y2");

years[2]=getParameter("y3");

years[3]=getParameter("y4");

years[4]=getParameter("y5");

for(int i=0;i<5;i++)

{

g.drawString(years[i],20,i\*50+30);

g.drawRect(50,i\*50+10,marks[i],30);

}

}

}

**//Exception Program**

**//Program to handle exception- arithmetic exception-divide by zero**

import java.lang.\*;

class Myexception

{

public static void main(String []args)

{

int a=10,b=0;

try

{ System.out.println("In try block");

int c=a/b;

System.out.println("division of a and b : "+c);

}

catch(Exception e)

{ System.out.println(e);

System.out.println("In catch block");

}

finally

{ System.out.println("In finally block");

System.out.println("Exception handled");

}

}

}

**//Write a program(WAP) to accept a password from the user and throw "authentication Failure" if the password is incorrect**.

import java.util.Scanner;

class PasswordCheck

{

public static void main(String [] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter ur password : ");

String pwd=sc.next();

try

{

if(pwd.equals("abc123"))

{

System.out.println("Authentication Success");

}

else

{

throw new passwordException("Authentication Failure");

}

}

catch(passwordException e)

{

System.out.println(e);

}

}

}

class passwordException extends Exception

{

passwordException(String str)

{

super(str);

}

}

**//write a program to create two threads, one to print numbers in orignal order and other in reverse order from 1 to 10.**

import java.lang.Thread.\*;

class Mythread extends Thread

{

public void run()

{

for(int i=1;i<=10;i++)

{

System.out.println("print 1 to 10 : "+i);

}

}

public static void main(String [] args)

{

Mythread mt=new Mythread();

Thread t1=new Thread(mt);

Mythread1 mt1=new Mythread1();

Thread t2=new Thread(mt1);

t1.start();

t2.start();

}

}

class Mythread1 extends Thread

{

public void run()

{

for(int i=10;i>=1;i--)

{

System.out.println("reverse 10 to 1 : "+i);

}

}

}

**//Exception example- ArrayIndexOutOfBond**

import java.lang.\*;

class Myexception2

{

public static void main(String [] args)

{

int arr[]=new int[5];

try

{

System.out.println("TRY");

arr[6]=10;

System.out.println("array : "+arr[6]);

}

catch(Exception e)

{

System.out.println("CATCH");

System.out.println(e);

}

finally

{ System.out.println("FINALLY");

System.out.println("Exception handled");

}

}

}

**//Strings methods :**

package demo;

import java.lang.\*;

class pkg\_string

{

public static void main(String []args)

{

String s1="Hello world";

String s2="World";

//1.CharAt() - finds character at specified location or index

char ch = (s1.charAt(2));

System.out.println(ch);

//2.substring : substring(starting index,end index)-finds substring of starting index //and ending-1

String sub=s1.substring(0,3);

System.out.println("Substring is "+sub);

//3.indexOf- find character of particular index

int i=s1.indexOf('o');

System.out.println("index is "+i);

//4.replace-replace string with another

String re=s2.replace("World","hello");

System.out.println("Replace : "+re);

//5.length

int l=s1.length();

System.out.println("length:"+l);

}

}

/\* -To compile package program we use -d or -classpath command,after that . indicates current directory or if u want to save .class file in other directory then specify directory name instead of .

g:\Java>javac -d . pkg\_string.java

-To run package programs fully qualified name is used i.e.packagename.classname

g:\Java>java demo.pkg\_string

o/p :

l

Substring is Hel index is 4

Replace : hello length:11

\*/

**//File Program - Write a program that will count number of characters in a file**

import java.io.\*;

class CountChar

{

public static void main(String [] args)

{

try{

//open file in read mode

FileReader fr=new FileReader("a.txt");

int ch;

int count=0;

//loop works until all character in file are read,-1 indicates end of file

while((ch=fr.read())!=-1)

{

count++;

}

fr.close();

System.out.println("Number of characters in file are :"+count);

}

catch(Exception e) { }

}

}

/\* output :

g:\Java>javac CountChar.java

g:\Java>java CountChar

Number of characters in file are :31

\*/

**//File Program - Write a program to copy contents of one file to another file.**

**// also Exception example- throws keyword**

import java.io.\*;

public class FileCopy

{

public static void main(string[] args) throws IOException

{ FileInputStream in=new FileInputStream("input.txt");

FileOutputStream out=new FileOutputStream("output.txt");

try

{

int ch=0;

while(ch!=-1)

{ ch=in.read();

out.write(ch);

}

System.out.println("File Copied..");

}

finally

{ in.close();

out.close();

}

}

}