Java Programming using Linux

Software Lab V

1. **Program to check whether a given number is palindrome, prime, or Armstrong mumber**

```
import java.util.*;
import java.io.*;
class Number
{
    int num;
    void getNumber(int n)
    {  num=n; }
    void palindrome()
    {
      int p,d;
      p=num;
      int rev=0;`
      while(p>0)
      {
          d=p%10;
          rev=d+rev*10;
          p=p/10;
               }
      if (rev==num)
       System.out.println("the num is palindrome");
      else
       System.out.println("the num is not a paindrome");
         }


void prime()
```

```
    {
        int count=0;
        for(int i=1;i<=num;i++)
            if((num%i)==0)
                count++;
        if (count>2)
        System.out.println(" the number is not prime");
        else
        System.out.println("the number is a prime number");
    }
    void armstrong()
    {
        int p,d;
      p=num;int s=0;
      while(p>0)
      {
          d=p%10;
          s=s+d*d*d;
          p=p/10;
  }
     if (s==num)
       System.out.println("the num is armstrong");
      else
       System.out.println("the num is not a armstrong");
    }}

    public class Main
{
          public static void main(String[] args)  {
            Scanner sc=new Scanner(System.in);
            Number numb=new Number();

            System.out.println("enter the number to be checked ");
            int n= sc.nextInt();
            numb.getNumber(n);
             int choice;
            do{
               System.out.println("enter your choice");
                choice=sc.nextInt();
               switch(choice)
               {
                  case 1:
                     numb.palindrome();
                     break;
                 case 2:
                    numb.prime();
                    break;
                 case 3:
                    numb.armstrong();
                    break;
                 case 4:
                    break;
               }
            }while(choice<4);

         }
   }
```

2. **Write a Program to implement Method Overloading**

```
import java.util.*;
class shape
{
   void area(int l,int b)
   {       System.out.println("area of the rectangle="+(l*b));    }
   void area(int side)
   {       System.out.println("area of square=" +(side*side));  }
      void area(float b,float h)
      {           System.out.println("area of triangle="+(0.5*b*h));          }
      void area(float r)
      {           System.out.println("area of circle="+(3.14*r*r));       }
}
public class methoverload
{
   public static void main(String arg[])
   {
      shape s=new shape();
      Scanner sc=new Scanner(System.in);
      System.out.println("rectangle");
      System.out.println("enter the length of the rectangle");
      int l=sc.nextInt();
      System.out.println("enter the breadth of the rectangle");
      int b=sc.nextInt();
      s.area(l,b);
      System.out.println("square");
      System.out.println("enter the side");
      int side=sc.nextInt();
      s.area(side);
      System.out.println("triangle");
      System.out.println("enter the base");
      float c=sc.nextFloat();
      System.out.println("enter the height");
      float a=sc.nextFloat();
      s.area(a,c);
      System.out.println("circle");
      System.out.println("enter the radious of circle");
      float r=sc.nextFloat();
      s.area(r);
   }
}
```

3. **Write a program to implement Constructor overloading**

```
import java.util.*;

class rect
{
   float l,b;

rect()
   {
      l=45.09f;                  b=69.20f;
   }
   rect(float x,float y)
   {
      l=x;        b=y;
   }
   rect(float x)
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("enter the length of the rectangle");
      l=sc.nextFloat();
      b=x;
   }
   void disp()
   {
      System.out.println("length="+l);
      System.out.println("breadth="+b);
      System.out.println("area="+(l*b));
   }
}

public class rectconstructor
{
   public static void main(String arg[])
   {
          Sytem.out.println(“ The first Rectangle”);
         rect R=new rect();
         R.disp();
           Sytem.out.println(“ The Second  Rectangle”);
           rect R1=new rect(3.4F,6.9F);
             R1.disp();
Sytem.out.println(“ The third  Rectangle”);
           rect R2=new rect(3.2f);
      R2.disp();
   }
}
```

4. **Create a class called Complex which contains two data members X and Y. Include the following member functions**

a) to read a complex number,
b) to display a complex number,
c) to add two complex numbers
d) to multiply two complex numbers.

```
import java.io.*;

class complex

{

   float real;

   float img;

   complex()

   {

      real=0.0f;       img=0.0f;

   }

   void read()throws IOException{

      DataInputStream in=new DataInputStream(System.in);

      System.out.println("Enter the real part");

      real=Float.valueOf(in.readLine());

      System.out.println("Enter the imaginary part");

      img=Float.valueOf(in.readLine());


   }

   void display()

   {      System.out.println(real+"+i" +img);         }

   void sum(complex c1,complex c2)

   {

      real=c1.real+c2.real;       img=c1.img+c2.img;

   }

   void mul(complex c1,complex c2)

   {

      real=c1.real*c2.real;       img=c1.img*c2.img;
   }
}
public class complexarithemetic{
   public static void main(String args[])throws IOException
   {
      complex C1=new complex();
      complex C2=new complex();
      complex res=new complex();
      System.out.println("Enter the first complex number");
      C1.read();
      System.out.println("Enter the second complex number");
      C2.read();
      System.out.println("first complex number");
      C1.display();
      System.out.println("second complex number");
      C2.display();
      res.sum(C1,C2);
   System.out.println("the resultant complex number is ");
   res.display();
   res.mul(C1,C2);
   System.out.println("the resultant complex number is");
   res.display();
   }
}
```

5. Create a class matrix which contains a 2d integer array, m & n as data member. Include the following member functions:

a. To read the matrix

b. To display the matrix

c. Display the transpose of the matrix

```
import java.io.*;
class matrix
{
int mat[][];
int row,col;
matrix(int r,int c){
row=r; col=c;
mat=new int[row][col];
}
void getdata() throws IOException
{
DataInputStream in=new DataInputStream(System.in);
System.out.println("Enter the elements in the array");
for (int i=0;i<row;i++)
for (int j=0;j<col;j++)
mat[i][j]=Integer.parseInt(in.readLine());
}
void display()
{

for (int i=0;i<row;i++)
{

        for (int j=0;j<col;j++)
{
        System.out.print(mat[i][j]+"\t");
}
System.out.println();
}
}

void transpose()
{
for(int i=0;i<row;i++)
{
for (int j=0; j<col;j++)
 {
System.out.print(mat[j][i]+"\t");
}
System.out.println();}
}
}

class matrixdemo{
public static void main(String args[])throws IOException
{
matrix A=new matrix(3,3);
matrix B=new matrix(2,2);
A.getdata();
System.out.println("The matrix is....");
A.display();
System.out.println("The transpose of the matrix is....");
A.transpose();
B.getdata();
System.out.println("The matrix is....");
B.display();
System.out.println("The transpose of the matrix is....");
```

```
B.transpose();
}
}
```

7. **Write a Program to implement multilevel inheritance**

```
import java.io.*;
class personal{
        String name;
        int rollno;
        void getpersonal() throws IOException
                {
                        DataInputStream in=new DataInputStream(System.in);
                        System.out.println(" enter the name of student");
                        name=in.readLine();
                        System.out.println(" enter the roll number of the student");
                        rollno=Integer.parseInt(in.readLine());
                }
        }
class test extends personal
        {
        int sub1,sub2,sub3;
        void getmarks() throws IOException
        {       DataInputStream in=new DataInputStream(System.in);

                getpersonal();
                System.out.println("Enter the marks:");

                sub1=Integer.parseInt(in.readLine());
                sub2=Integer.parseInt(in.readLine());
                sub3=Integer.parseInt(in.readLine());
        }
}

class student extends test
        {
        void display()
        {
        int total=sub1+sub2+sub3;
        int per=(int)(total/3);
        System.out.println(" Name :" +name);
        System.out.println(" Roll No: "+ rollno);
        System.out.println(" marks : "+ sub1+"\t"+sub2+"\t"+sub3);
        System.out.println(" Total Marks: "+ total);
        System.out.println(" percentage of Marks: "+ per+"%");
        if(per>90)
        System.out.println(" Grade:A+ ");
        else if(per>80)
        System.out.println(" Grade:A ");
        else if(per>60)
        System.out.println(" Grade:B ");
        else if(per>50)
        System.out.println(" Grade:C ");
        else
        System.out.println(" Grade: D");

        }}

 class multilevel
```

```
{
public static void main(String arg[])throws IOException
{
student s=new student();
System.out.println(" enter the details of student\n\n****************");
s.getmarks();
s.display();
}
}
```

8. The program for implementing interface

```
import java.io.*;
class student
{
String name;
int roll;
void getroll() throws IOException
{
DataInputStream in=new DataInputStream(System.in);
System.out.println("Enter the name and roll number of the student");
name=in.readLine();
roll=Integer.parseInt(in.readLine());
}
}
class test extends student
{
 int m1,m2,m3;
 void getmarks() throws IOException
 {
DataInputStream in=new DataInputStream(System.in);
System.out.println("Enter the marks of the student");

m1=Integer.parseInt(in.readLine());
m2=Integer.parseInt(in.readLine());
m3=Integer.parseInt(in.readLine());
}}
interface sports{
final int wt=5;
}

class result extends test implements sports
{
int total;
void getdetails() throws IOException
{
getroll();
getmarks();
}
void display()
{
 total= m1+m2+m3+wt;
System.out.println("Roll Number:\t"+ roll);
System.out.println("Name      :\t"+ name);
System.out.println("Total     :\t"+ total);
}
}
class studentdetails{
```

```
public static void main(String arg[]) throws IOException

{

result R= new result();

R.getdetails();

System.out.println("The details of the student\n ******************");

R.display();

}}
```

9. **Program to implement super keyword**

```
//program to implement the uses of super keyword.

import java.io.*;

class A
{
int i,j;
A()
{}
A(int x,int y)// base class constructor
{
i=x;j=y;
}
}

class B extends A{
int i;
B(){}
B(int p,int q,int r)
{
super(p,q);// invoking the base class constructor
i=r;
}
void getdata(int i)
{
super.i=34;
super.j=25;
this.i=i; }
void display(){
System.out.println("i in superclass="+super.i);
System.out.println("j ="+j);
System.out.println("i in subclass ="+i);
}
}
 public class Superkey{
public static void main(String args[]){
B obj1=new B(2,4,6);
B obj2=new B();
System.out.println(" The First object\n************************");
obj1.display();
System.out.println("The Second Object\n***********************");
obj2.getdata(37);
obj2.display();
}}
```

10. Write a java program to implement Method overriding. (resolve using super keyword)

```
class A {
int i, j;
A(int a, int b) {
i = a;
j = b;
}
// display i and j
void show() {
System.out.println("i and j: " + i + " " + j);
}
}

class B extends A {

int k;
B(int a, int b, int c) {
super(a, b);// calls constructor of base class
k = c;
}
// display k – this overrides show() in A
void show() {
super.show();//this calls the show() in A
System.out.println("k: " + k);
}
}
class Override {
public static void main(String args[]) {
B subOb = new B(1, 2, 3);
subOb.show(); // this calls show() in B
}
}
```

11. **Write a program to implement dynamic method dispatch**

```
class A {
void callme() {
System.out.println("Inside A's callme method");
}
}
class B extends A {
// override callme()
void callme() {
System.out.println("Inside B's callme method");
}
}
class C extends A {
// override callme()
void callme() {
System.out.println("Inside C's callme method");
}
}
class Dispatch {
public static void main(String args[]) {
A a = new A(); // object of type A
B b = new B(); // object of type B
C c = new C(); // object of type C
A r; // obtain a reference of type A
r = a; // r refers to an A object
r.callme(); // calls A's version of callme
r = b; // r refers to a B object
r.callme(); // calls B's version of callme
r = c; // r refers to a C object
r.callme(); // calls C's version of callme
}
}
```

12. **Define an abstract class Geometry that contains a member variable of type int. Declare an abstract method findArea(). Derive classes Rectangle, Triangle and Circle from Geometry class. Override method findArea() to print the area of the following:**

1) **Rectangle 2) Triangle 3) Circle.**

```
// Using abstract methods and classes.

abstract class Geometry {

int  dim1;

Geometry(int a)

 {

dim1 = a;
```

```
}
// findArea() is now an abstract method
abstract double findArea();
}
class Rectangle extends Geometry {
int b;
Rectangle(int x, int  y) {
super(x);
b=y;
}
// override area for rectangle
int findArea() {
System.out.println("Inside Area for Rectangle.");
return dim1 * b;
}
}
class Triangle extends Geometry {
int h;
Triangle(int  x, int y) {
super(x);
h=y;
}
// override area for right triangle
double findArea() {
System.out.println("Inside Area for Triangle.");
return dim1 * h / 2;
}
class Circle extends Geometry{
Circle(  int r)
```

```
{ super(r);
}
double findArea()
{
System.out.println(" The Area of circle")'
return  (3.14 f* dim1*dim1);
}}
class GeometryMain{
public static void main(String args[]) {
Rectangle r = new Rectangle(9, 5);
Triangle t = new Triangle(10, 8);
Circle c=new Circle(4);
Geometry g;
g=r;
System.out.println("Area is " + g.findArea());
g=t;
System.out.println("Area is " + g.findArea());
g=c;
System.out.println("Area is " + g.findArea());
}
}
```

13. Write a Java program to create a class Complex for reading and displaying a complexnumber under a user defined package complex.

procedure:

- **Create folder complex**
- **Create the file Complex.java**
- **Save this file in the complex folder**
- **Compile the Complex.java program within the complex folder**
- **import the package complex and use the class Complex in Complexmain.java file**

Complex.java

```
package complex;
public class Complex{
int real,img;
public void read(int x,int y)
{
real=x;
img=y;
}
public void display()
{
System.out.println("the Complex number is: "+ real + "+i"+ img);
}
}
```

Complexmain.java

```
import complex.Complex;

class Complexmain{

public static void main(String args[])

{

Complex C=new Complex();

C.get(2,4);

C.display();

}}
```

14. **Write a Java Program to print factorial of a given number. If the number is negative then throw a user defined exception.**

```
import java.io.*;
class negativeexception extends Exception{
negativeexception(){ }
public String toString(){
return "You entered a negative number ";
}
}
class Factorial{
static void factorial(int a)  throws negativeexception{
if (a<0)
throw new negativeexception();
else{
int f=1;
for(int i=1;i<=a;i++)
f*=i;
System.out.println("Factorial of"+a+"="+f);
}
}
public static void main(String arg[]) throws IOException{

try{
```

```
int x,y;
DataInputStream in=new DataInputStream(System.in);
System.out.println(" enter the number");
x=Integer.parseInt(in.readLine());
factorial(x);
}
catch(negativeexception e){
System.out.println(e);
}
}}
```

15. **Write a Java program to implement two threads one for printing odd numbers and another for printing even numbers simultaneously up to a given number.**

```
import java.io.*;
class A extends Thread
{
int limit;
A(int n)
{ limit=n;}

public void run()
{
for (int i=1;i<=limit;i++)

{ if (i%2==1)
System.out.println ("\t odd numbers : i="+i);  }
}
}

class B extends Thread
{
int limit;
B(int n)
{ limit=n;}
public void run()
{
for (int j=1;j<=limit;j++)

{
if (j%2==0)
System.out.println ("\t even numbers: j="+j);  }
}
}

class mainthread
```

```
{
public static void main(String args[]) throws IOException
{
DataInputStream in=new DataInputStream(System.in);
int limit=Integer.parseInt(in.readLine());
A t1=new A(limit); t1.start();
B t2=new B(limit);  t2.start();
}}
```

APPLET PROGRAMS

16. Write an applet program to display a human face

```
import java.awt.*;
 import java.applet.*;

/*<applet code=face.class width=500 height=300>
</applet>
*/

public class face extends Applet
{
        public void paint(Graphics g)
                {
                g.drawOval(40,40,120,150);
                g.drawOval(57,75,30,20);
                g.drawOval(110,75,30,20);
                g.fillOval(68,81,10,10);
                g.fillOval(121,81,10,10);
                g.drawOval(85,100,30,30);
                g.fillArc(60,125,80,40,180,180);
                g.drawOval(25,92,15,30);
                g.drawOval(160,92,15,30);
                }
}
```

17. Write an applet program to display the National Flag

```
import java.awt.*;
 import java.applet.*;

/*<applet code=flag.class width=500 height=300>
</applet>*/
```

```
public class flag extends Applet
{
        public void paint(Graphics g)
        {
        g.setColor(Color.orange);
        g.fillRect(50,80,200,40);
        g.setColor(Color.white);
        g.fillRect(50,120,200,40);
        g.setColor(Color.green);
        g.fillRect(50,160,200,40);
        g.setColor(Color.black);
        g.fillRect(40,80,10,320);
        g.drawOval(140,130,25,25);
        }
}
```

18. Write an applet program to display Olympic ring.

```
import java.awt.*;
 import java.applet.*;

/*<applet code=ring.class width=600 height=600>
</applet>
*/

public class ring extends Applet
{
public void paint(Graphics g)
{
           g.setColor(Color.blue);
           g.drawOval(140,130,100,100);
           g.setColor(Color.yellow);
           g.drawOval(200,180,100,100);
           g.setColor(Color.black);
           g.drawOval(260,130,100,100);
           g.setColor(Color.green);
           g.drawOval(320,180,100,100);
           g.setColor(Color.red);
           g.drawOval(380,130,100,100);
}}
```

19. Write a program to display a moving banner

```
      import java.awt.*;
      import java.applet.*;

      /* <applet code="banner.class" height=400 width=400>
```

```
</applet>
*/

public class banner extends Applet implements Runnable
{
String msg="HAPPY BIRTHDAY TO YOU........";
Thread t=null;
boolean stopFlag;

public void init()//applet class
{  setBackground(Color.cyan);
setForeground(Color.red);       }

public void start()//runnable
{    t=new Thread(this);
stopFlag=false;
t.start();    }

public void run()//thread class
{
char ch;
for(;;)//infinite loop
{
try {
repaint();
Thread.sleep(500);
ch=msg.charAt(0);
msg=msg.substring(1,msg.length());
msg+=ch;
if(stopFlag)
break;      }
catch(InterruptedException e)  {          }
}    }

public void stop()
{    stopFlag=false;
t=null;      }

public void paint(Graphics g){
Font f= new Font("Times New Roman", Font.BOLD|Font.ITALIC,26);
g.setFont(f);
g.drawString(msg,30,100);}
}
```

20. Write an applet program to display a house using drawpolygon method.

```
import java.awt.*;
 import java.applet.*;
```

```
/*<applet code=house.class width=500 height=300>
</applet>
*/

public class house extends Applet{

public void paint(Graphics g)
{
int x[]={50,110,80,50};
int y[]={80,80,20,80};
g.drawPolygon(x,y,3);
g.drawRect(50,80,60,70);
g.drawRect(75,110,15,35);
}}
```

21. **Write a GUI program to find sum of two numbers**

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

class fra extends JFrame implements ActionListener{
JButton b;;JTextField t1,t2,t3;
Container con;
fra(String s)
{super(s);
setSize(300,300);
con=getContentPane();
con.setLayout(new FlowLayout(FlowLayout.LEFT));
b=new JButton("add");
JLabel l1=new JLabel("First",10);
t1=new JTextField(20);
JLabel l2=new JLabel("second",10);
t2=new JTextField(20);
JLabel l3=new JLabel("result",10);
t3=new JTextField(10);
b.addActionListener(this);
con.add(l1);
con.add(t1);
con.add(l2);
con.add(t2);
con.add(l3);
con.add(t3);
con.add(b);
addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
}
public void actionPerformed(ActionEvent ae)
{
int a=Integer.parseInt(t1.getText());
int b=Integer.parseInt(t2.getText());
int c=a+b;
t3.setText(""+c);
}
}

class guisum{
public static void main(String arg[])
{
JFrame f=new fra("skjdjk");
f.setVisible(true);

}}
```

22. Write a java GUI program to implement simple interest calculator (I=P*N*R).

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

class fra extends JFrame implements ActionListener
{
JButton b;;JTextField t1,t2,t3,t4;
Container con;
fra(String s)
{
super(s);
setSize(250,300);
con=getContentPane();
con.setLayout(new FlowLayout(FlowLayout.LEFT));
b=new JButton("calculate");
JLabel l1=new JLabel("principal amount");
t1=new JTextField(10);
JLabel l2=new JLabel("rate of interest");
t2=new JTextField(20);
JLabel l3=new JLabel("no. of years");
t3=new JTextField(20);
JLabel l4=new JLabel("interest     ",10);
t4=new JTextField(20);
b.addActionListener(this);

con.add(l1);con.add(t1);
con.add(l2);con.add(t2);
con.add(l3);con.add(t3);
con.add(l4);con.add(t4);
con.add(b);

addWindowListener(new WindowAdapter()
{
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});

}
public void actionPerformed(ActionEvent ae)
{
float p=Float.valueOf(t1.getText());
float r=Float.valueOf(t2.getText());
int n =Integer.parseInt(t3.getText());
float I=p*n*r/100;
t4.setText(""+I);
}
}

class simpleinterest{
public static void main(String arg[])
{
JFrame f=new fra("Simple Interest");
f.setVisible(true);

}}
```