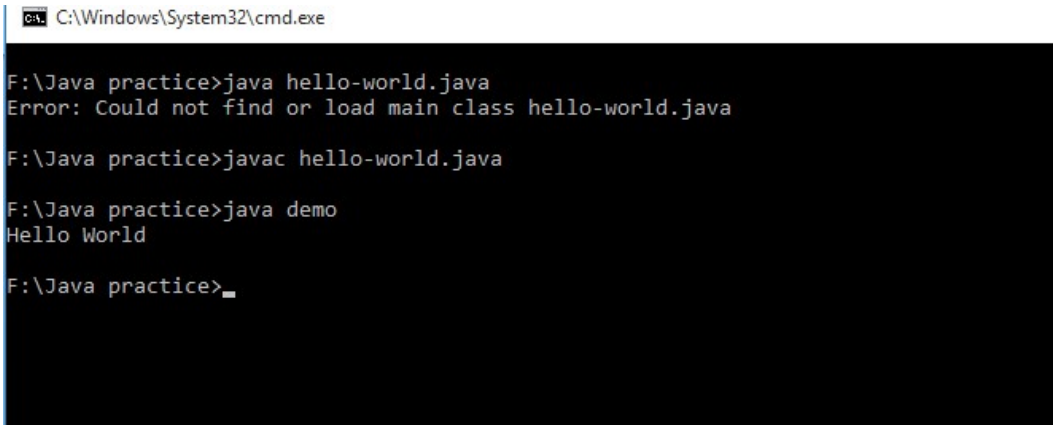
P-1 WAP to print some statements like “Hello World!”.

```
class demo{

public static void main(String args[])

        {

        System.out.println("Hello World");        }

}
```



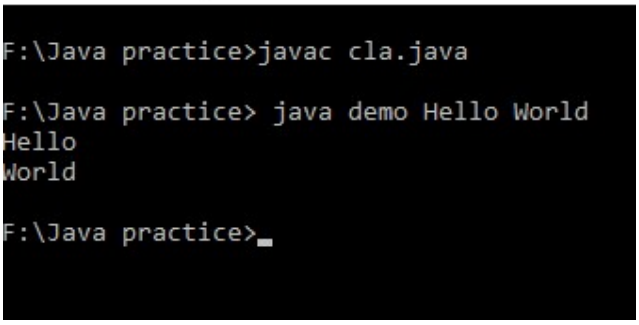
P- 2 WAP to calculate room area using multiple classes.

```
import java.util.Scanner;
class demo
{
  public static void main(String args[])
   {

    Scanner s= new Scanner(System.in);

System.out.println("Enter the length:");
      double l= s.nextDouble();
System.out.println("Enter the breadth:");
      double b= s.nextDouble();


double  area=l*b;
System.out.println("Area of Rectangle is: " + area);
  }
}
```

```
F:\Java practice>javac room.java

F:\Java practice>java demo
Enter the length:
10
Enter the breadth:
10
Area of Rectangle is: 100.0

F:\Java practice>
```

P-3 WAP to demonstrate the use of command line arguments.

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

for(int i=0;i<args.length;i++)
System.out.println(args[i]);

}
}
```



P-4. WAP to explain the basic data types used in java.

```
class demo {
	public static void main(String args[])
	{
		char a = 'G';
		int i = 89;
		byte b = 4;
		short s = 56;
		double d = 4.355453532;
		float f = 4.7333434f;

		System.out.println("char: " + a);
		System.out.println("integer: " + i);
		System.out.println("byte: " + b);
		System.out.println("short: " + s);
		System.out.println("float: " + f);
		System.out.println("double: " + d);
	}
}
```

```
F:\Java practice>javac basic-datatype.java

F:\Java practice>java demo
char: G
integer: 89
byte: 4
short: 56
float: 4.7333436
double: 4.355453532
```

P-5. WAP to explain the type casting in java.

Implicit Typecasting :

```
class demo {
  public static void main(String[] args) {
    int myInt = 9;
    double myDouble = myInt;

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

```
F:\Java practice>javac implicit.java

F:\Java practice>java demo
9
9.0

F:\Java practice>
```

Explicit Typecasting:

```
class demo {
  public static void main(String[] args) {
    double myDouble = 9.78;
    int myInt = (int) myDouble;

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

```
F:\Java practice>javac explicit.java

F:\Java practice>javac explicit.java

F:\Java practice>java demo
9.78
9

F:\Java practice>
```

P-6 WAP to demonstrate java expressions using different operators in java like relational, logical, bitwise operators etc.

Unary Operator:

```
class demo{

public static void main(String args[]){

int x=10;

System.out.println(x++);

System.out.println(++x);

System.out.println(x--);

System.out.println(--x);

}

}
```

```
F:\Java practice>javac operator.java

F:\Java practice>java demo
10
12
12
10
```

Arithmetic Operator:

```
class demo{

public static void main(String args[]){

int a=10;

int b=5;

System.out.println(a+b);

System.out.println(a-b);

System.out.println(a*b);
```

```
System.out.println(a/b);

System.out.println(a%b);

}}
```

```
F:\Java practice>javac operator.java

F:\Java practice>java demo
15
5
50
2
0
```

Java Left Shift Operator:

```
class demo{

public static void main(String args[]){

System.out.println(10<<2);

System.out.println(10<<3);

System.out.println(20<<2);

System.out.println(15<<4);

}}
```

```
F:\Java practice>javac operator.java

F:\Java practice>java demo
40
80
80
240
```

Right Shift Operator:

```
class demo{

public static void main(String args[]){

System.out.println(10>>2);

System.out.println(20>>2);

System.out.println(20>>3);

}}
```

```
F:\Java practice>javac operator.java

F:\Java practice>java demo
2
5
2
```

AND Operator Example: Logical && and Bitwise &:

```
class demo{

public static void main(String args[]){

int a=10;

int b=5;

int c=20;

System.out.println(a<b&&a<c);

System.out.println(a<b&a<c);

}}
```

```
F:\Java practice>javac operator.java

F:\Java practice>java demo
false
false
```

OR Operator Example: Logical || and Bitwise | :

```
class demo{

public static void main(String args[]){

int a=10;

int b=5;

int c=20;

System.out.println(a>b||a<c);

System.out.println(a>b|a<c);

System.out.println(a>b||a++<c);

System.out.println(a);

System.out.println(a>b|a++<c);

System.out.println(a);

}}
```

```
F:\Java practice>javac operator.java

F:\Java practice>java demo
true
true
true
10
true
11
```

Ternary Operator:

```
class demo{

public static void main(String args[]){

int a=2;

int b=5;

int min=(a<b)?a:b;

System.out.println(min);

}}
```

```
F:\Java practice>javac operator.java

F:\Java practice>java demo
2
```

Assignment Operator:

```
class demo{

public static void main(String args[]){

int a=10;

int b=20;

a+=4;

b-=4;

System.out.println(a);

System.out.println(b);

}}
```

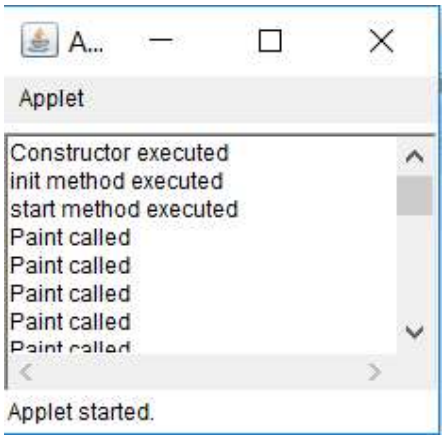
```
F:\Java practice>javac operator.java

F:\Java practice>java demo
14
16
```

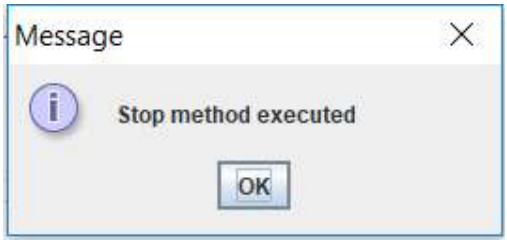
P-7. WAP to demonstrate if-else, nested if-else, if-else ladder.

```
publicclass Condition {
	publicstaticvoidmain(String[] args)
	{
	ifElse(10);
	nestedIfElse(12);
	ifElseLadder(1000);

	}

	// If Else
	publicstaticvoidifElse(int a)
	{
		if (a<5)
		{
			System.out.println("Value is below 5 i.e." + a);
		}
		else
		{
			System.out.println("Value is greater than 5 i.e." +a );
		}
	}

	// nested if-else
	publicstaticvoidnestedIfElse(int a)
	{
		if (a<10) {
			if(a<5)
			{
			    System.out.println("Value is less than 5 i.e. "+a);
			}
			else
			{
			    System.out.println("Value is greater than 5 i.e."+a);
			}
		}
		else
```

P 34- WAP to demonstrate Applet with all the states used in it.

```
package java_program;
import java.awt.*;
import java.applet.Applet;
import javax.swing.JOptionPane;

public class AppletLifecycle extends Applet {

    TextArea messages = new TextArea(8, 30);

    public AppletLifecycle() {
        add(messages);
        messages.append("Constructor executed\n");
    }
        public void init() {
        messages.append("init method executed\n");
    }
        public void start() {
        messages.append("start method executed\n");
    }
       public void paint(Graphics g) {
        messages.append("Paint called\n");
    }


      public void stop() {
        JOptionPane.showMessageDialog(null, "Stop method executed");
    }

    public void destroy() {
        JOptionPane.showMessageDialog(null, "Destory method executed");
    }

}
```





P-35: WAP to make graphic calculator.

```
package java_program;
import java.awt.*;
import java.awt.event.*;

public class MyCalculator extends Frame
{

public boolean setClear=true;
double number, memValue;
char op;

String digitButtonText[] = {"7", "8", "9", "4", "5", "6", "1", "2", "3", "0",
"+/-", "." };
String operatorButtonText[] = {"/", "sqrt", "*", "%", "-", "1/X", "+", "=" };
String memoryButtonText[] = {"MC", "MR", "MS", "M+" };
String specialButtonText[] = {"Backspc", "C", "CE" };

MyDigitButton digitButton[]=new MyDigitButton[digitButtonText.length];
MyOperatorButton operatorButton[]=new
MyOperatorButton[operatorButtonText.length];
MyMemoryButton memoryButton[]=new MyMemoryButton[memoryButtonText.length];
MySpecialButton specialButton[]=new MySpecialButton[specialButtonText.length];

Label displayLabel=new Label("0",Label.RIGHT);
Label memLabel=new Label(" ",Label.RIGHT);

final int FRAME_WIDTH=325,FRAME_HEIGHT=325;
final int HEIGHT=30, WIDTH=30, H_SPACE=10,V_SPACE=10;
final int TOPX=30, TOPY=50;

MyCalculator(String frameText)//constructor
{
super(frameText);

int tempX=TOPX, y=TOPY;
displayLabel.setBounds(tempX,y,240,HEIGHT);
displayLabel.setBackground(Color.RED);
displayLabel.setForeground(Color.WHITE);
```

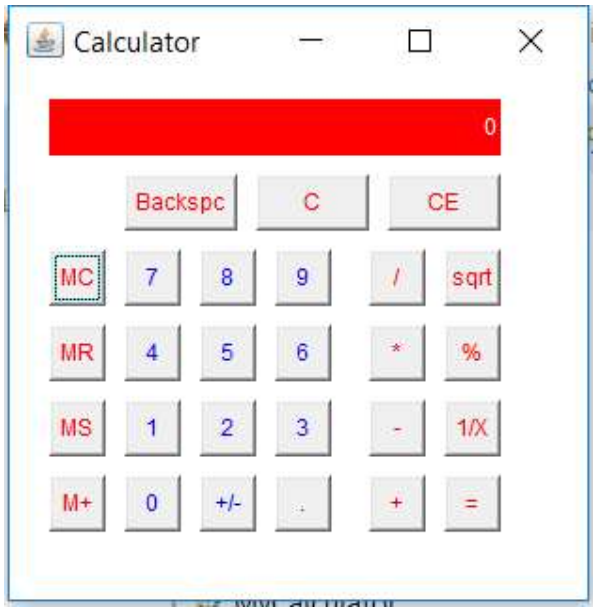
```
add(displayLabel);

memLabel.setBounds(TOPX,  TOPY+HEIGHT+ V_SPACE,WIDTH, HEIGHT);
add(memLabel);

// set Co-ordinates for Memory Buttons
tempX=TOPX;
y=TOPY+2*(HEIGHT+V_SPACE);
for(int i=0; i<memoryButton.length; i++)
{
memoryButton[i]=new MyMemoryButton(tempX,y,WIDTH,HEIGHT,memoryButtonText[i],
this);
memoryButton[i].setForeground(Color.RED);
y+=HEIGHT+V_SPACE;
}

//set Co-ordinates for Special Buttons
tempX=TOPX+1*(WIDTH+H_SPACE); y=TOPY+1*(HEIGHT+V_SPACE);
for(int i=0;i<specialButton.length;i++)
{
specialButton[i]=new
MySpecialButton(tempX,y,WIDTH*2,HEIGHT,specialButtonText[i], this);
specialButton[i].setForeground(Color.RED);
tempX=tempX+2*WIDTH+H_SPACE;
}

//set Co-ordinates for Digit Buttons
int digitX=TOPX+WIDTH+H_SPACE;
int digitY=TOPY+2*(HEIGHT+V_SPACE);
tempX=digitX;  y=digitY;
for(int i=0;i<digitButton.length;i++)
{
digitButton[i]=new MyDigitButton(tempX,y,WIDTH,HEIGHT,digitButtonText[i],
this);
digitButton[i].setForeground(Color.BLUE);
tempX+=WIDTH+H_SPACE;
if((i+1)%3==0){tempX=digitX; y+=HEIGHT+V_SPACE;}
}

//set Co-ordinates for Operator Buttons
int opsX=digitX+2*(WIDTH+H_SPACE)+H_SPACE;
int opsY=digitY;
tempX=opsX;  y=opsY;
for(int i=0;i<operatorButton.length;i++)
{
tempX+=WIDTH+H_SPACE;
operatorButton[i]=new
MyOperatorButton(tempX,y,WIDTH,HEIGHT,operatorButtonText[i], this);
operatorButton[i].setForeground(Color.RED);
if((i+1)%2==0){tempX=opsX; y+=HEIGHT+V_SPACE;}
}

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

setLayout(null);
```

```
setSize(FRAME_WIDTH,FRAME_HEIGHT);
setVisible(true);
}

static String getFormattedText(double temp)
{
String resText=""+temp;
if(resText.lastIndexOf(".0")>0)
      resText=resText.substring(0,resText.length()-2);
return resText;
}

public static void main(String []args)
{
new MyCalculator("Calculator");
}
}

class MyDigitButton extends Button implements ActionListener
{
MyCalculator cl;
MyDigitButton(int x,int y, int width,int height,String cap, MyCalculator clc)
{
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}
static boolean isInString(String s, char ch)
{
for(int i=0; i<s.length();i++) if(s.charAt(i)==ch) return true;
return false;
}
public void actionPerformed(ActionEvent ev)
{
String tempText=((MyDigitButton)ev.getSource()).getLabel();

if(tempText.equals("."))
{
 if(cl.setClear)
      {cl.displayLabel.setText("0.");cl.setClear=false;}
 else if(!isInString(cl.displayLabel.getText(),'.'))
      cl.displayLabel.setText(cl.displayLabel.getText()+".");
 return;
}

int index=0;
try{
        index=Integer.parseInt(tempText);
    }catch(NumberFormatException e){return;}

if (index==0 && cl.displayLabel.getText().equals("0")) return;

if(cl.setClear)
           {cl.displayLabel.setText(""+index);cl.setClear=false;}
else
      cl.displayLabel.setText(cl.displayLabel.getText()+index);
}//actionPerformed
```

```
}//class defination

class MyOperatorButton extends Button implements ActionListener
{
MyCalculator cl;

MyOperatorButton(int x,int y, int width,int height,String cap, MyCalculator
clc)
{
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}
public void actionPerformed(ActionEvent ev)
{
String opText=((MyOperatorButton)ev.getSource()).getLabel();

cl.setClear=true;
double temp=Double.parseDouble(cl.displayLabel.getText());

if(opText.equals("1/x"))
      {
      try
            {double tempd=1/(double)temp;
            cl.displayLabel.setText(MyCalculator.getFormattedText(tempd));}
      catch(ArithmeticException excp)
                  {cl.displayLabel.setText("Divide by 0.");}
      return;
      }
if(opText.equals("sqrt"))
      {
      try
            {double tempd=Math.sqrt(temp);
            cl.displayLabel.setText(MyCalculator.getFormattedText(tempd));}
            catch(ArithmeticException excp)
                  {cl.displayLabel.setText("Divide by 0.");}
      return;
      }
if(!opText.equals("="))
      {
      cl.number=temp;
      cl.op=opText.charAt(0);
      return;
      }
// process = button pressed
switch(cl.op)
{
case '+':
      temp+=cl.number;break;
case '-':
      temp=cl.number-temp;break;
case '*':
      temp*=cl.number;break;
case '%':
      try{temp=cl.number%temp;}
      catch(ArithmeticException excp)
            {cl.displayLabel.setText("Divide by 0."); return;}
```

```
      break;
case '/':
      try{temp=cl.number/temp;}
         catch(ArithmeticException excp)
                  {cl.displayLabel.setText("Divide by 0."); return;}
      break;
}//switch

cl.displayLabel.setText(MyCalculator.getFormattedText(temp));
//cl.number=temp;
}//actionPerformed
}//class

class MyMemoryButton extends Button implements ActionListener
{
MyCalculator cl;

MyMemoryButton(int x,int y, int width,int height,String cap, MyCalculator clc)
{
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}
public void actionPerformed(ActionEvent ev)
{
char memop=((MyMemoryButton)ev.getSource()).getLabel().charAt(1);

cl.setClear=true;
double temp=Double.parseDouble(cl.displayLabel.getText());

switch(memop)
{
case 'C':
      cl.memLabel.setText(" ");cl.memValue=0.0;break;
case 'R':
      cl.displayLabel.setText(MyCalculator.getFormattedText(cl.memValue));break
;
case 'S':
      cl.memValue=0.0;
case '+':
      cl.memValue+=Double.parseDouble(cl.displayLabel.getText());
      if(cl.displayLabel.getText().equals("0") ||
cl.displayLabel.getText().equals("0.0")  )
            cl.memLabel.setText(" ");
      else
            cl.memLabel.setText("M");
      break;
}//switch
}//actionPerformed
}//class

class MySpecialButton extends Button implements ActionListener
{
MyCalculator cl;

MySpecialButton(int x,int y, int width,int height,String cap, MyCalculator clc)
{
```

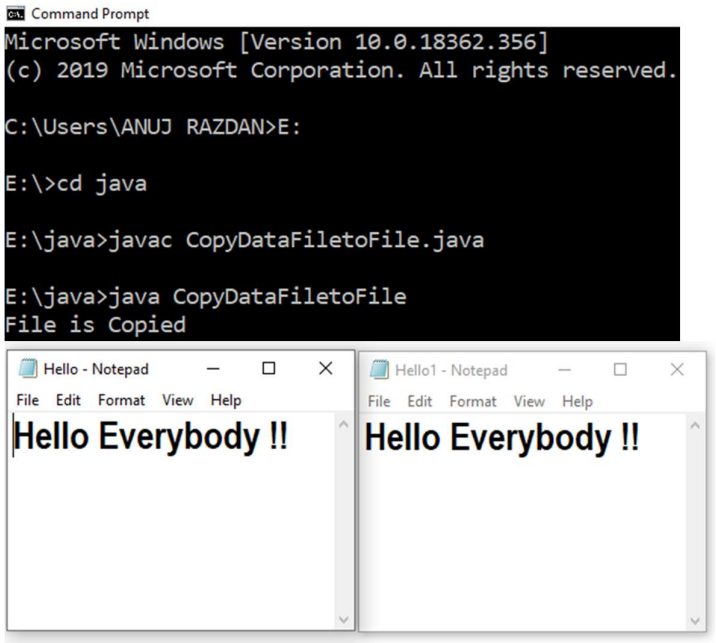
```
super(cap);
setBounds(x,y,width,height);
this.cl=clc;
this.cl.add(this);
addActionListener(this);
}
static String backSpace(String s)
{
String Res="";
for(int i=0; i<s.length()-1; i++) Res+=s.charAt(i);
return Res;
}
public void actionPerformed(ActionEvent ev)
{
String opText=((MySpecialButton)ev.getSource()).getLabel();
//check for backspace button
if(opText.equals("Backspc"))
{
String tempText=backSpace(cl.displayLabel.getText());
if(tempText.equals(""))
      cl.displayLabel.setText("0");
else
      cl.displayLabel.setText(tempText);
return;
}
//check for "C" button i.e. Reset
if(opText.equals("C"))
{
cl.number=0.0; cl.op=' '; cl.memValue=0.0;
cl.memLabel.setText(" ");
}

//it must be CE button pressed
cl.displayLabel.setText("0");cl.setClear=true;
}//actionPerformed
}//class
```



P 37- Write two different programs to copy a file in another file by character by character and byte by byte methods.

```
import java.io.*;
class CopyDataFiletoFile
{
        public static void main(String args[])throws IOException
    {
        FileInputStream Fread =new FileInputStream("Hello.txt");
        FileOutputStream Fwrite=new FileOutputStream("Hello1.txt") ;
        System.out.println("File is Copied");
        int c;
        while((c=Fread.read())!=-1)
        Fwrite.write((char)c);
        Fread.close();
        Fwrite.close();
    }
}
```

OUTPUT:-

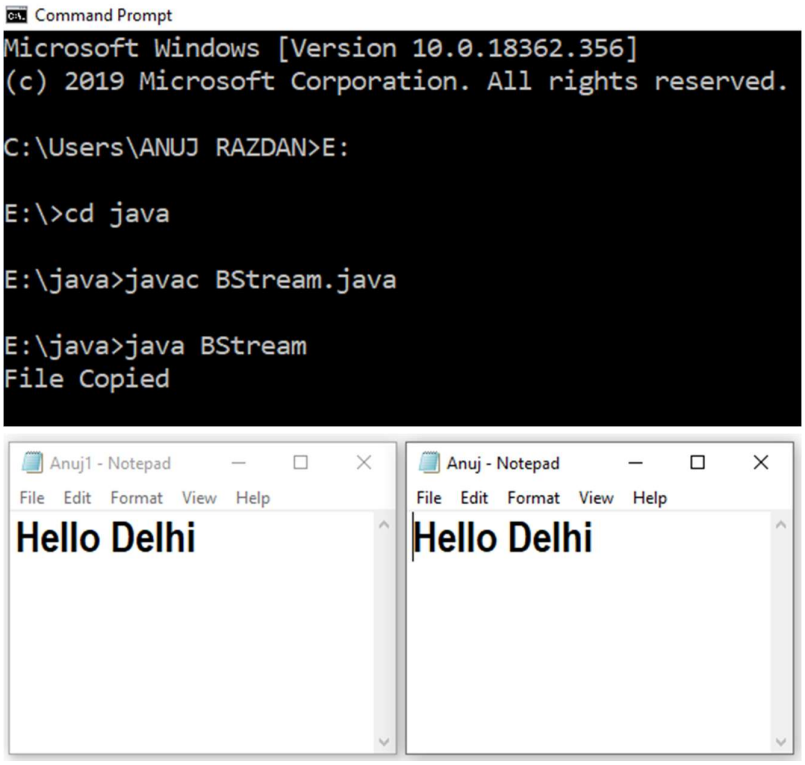


// Copy a File in Another File by Byte by Byte method

```
import java.io.*;
public class BStream
{
   public static void main(String[] args) throws IOException
   {
      FileInputStream  Dread = null;
      FileOutputStream Dwrite = null;

      try
      {
         Dread = new FileInputStream("Anuj.txt");
         Dwrite = new FileOutputStream ("Anuj1.txt");

         // Reading source file and writing content to target
         // file byte by byte
         int temp;
         while ((temp = Dread.read()) != -1)
            Dwrite.write((byte)temp);
                                    System.out.println("File Copied");
      }
      finally
      {
         if (Dread != null)
            Dread.close();
         if (Dwrite != null)
            Dwrite.close();
```

```
        }
    }
}
```

OUTPUT::-

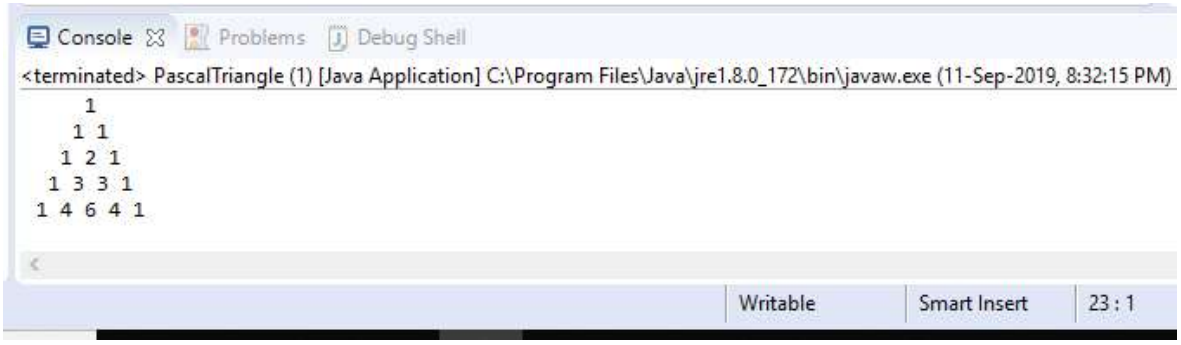


P 37- Write a Java application to print Pascal's triangle

```
class PascalTriangle
{
    public static void main(String args[])
    {
        int i,r=5,number,k,j;
        // r is number of rows
        for(i=0; i<r; i++)
        {
            for(k=r; k>i; k--)
            {
                System.out.print(" ");
            }
            number = 1;
            for(j=0; j<=i; j++)
            {
```

```
                    System.out.print(number+ " ");
                    number = number * (i-j) / (j+1);
                }
                System.out.println();
            }
        }
}
```

OUTPUT-

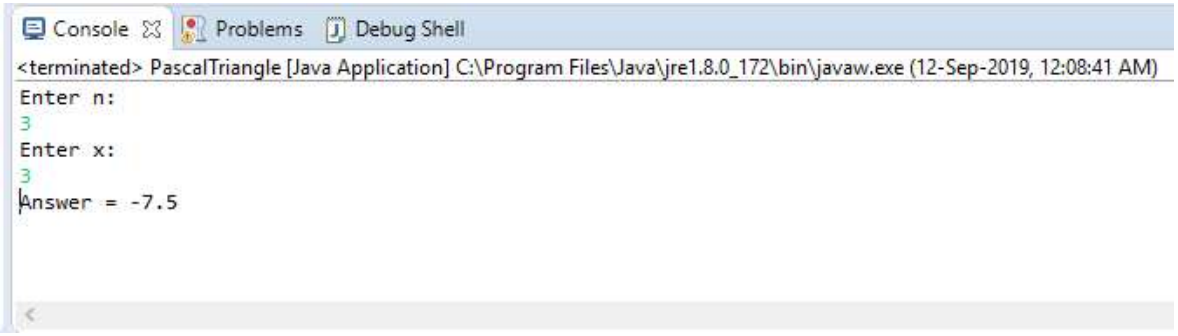


P 38- Write a Java program to compute the following series-
1 – x + x^2/2! – x^3/3! + x^4/4!....+(-/+1)nx^n/n! where x and n is accepted by user

```
import java.util.Scanner;
class FactorialSeries
{
        public static int fact(int index)
        {
            int f=1,i;
            for(i=1; i<=index; i++)
            {
                // to find factorial
                f = f*i;
            }
            return f;
```

```
        }
        publicstaticvoidmain(String[] args)
        {
                intn,x,i,num=-1;
                doublefrac,sum = 0;
                Scanner sc = newScanner(System.in);
                System.out.println("Enter n: ");
                n  =sc.nextInt();
                System.out.println("Enter x: ");
                x = sc.nextInt();
                for(i=1; i<=n; i++)
                {
                        frac = Math.pow(num,i)*((i*Math.pow(x,i))/fact(i));
                        sum = sum + frac;
                }
                System.out.println("Answer = " +sum);
        }
}
```

OUTPUT-



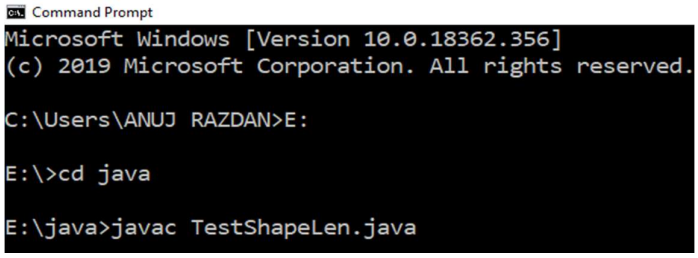
P 40- Create an abstract class Shape and derived classes Rectangle and Circle from Shape class. Implement abstract method of shape class in Rectangle and Circle class. Shape class contains: origin (x,y) as data member, display() and area() as abstract methods. Circle class contains: radius as data member. Rectangle class contains: length and width. (Use Inheritance, overloading and overriding concept)

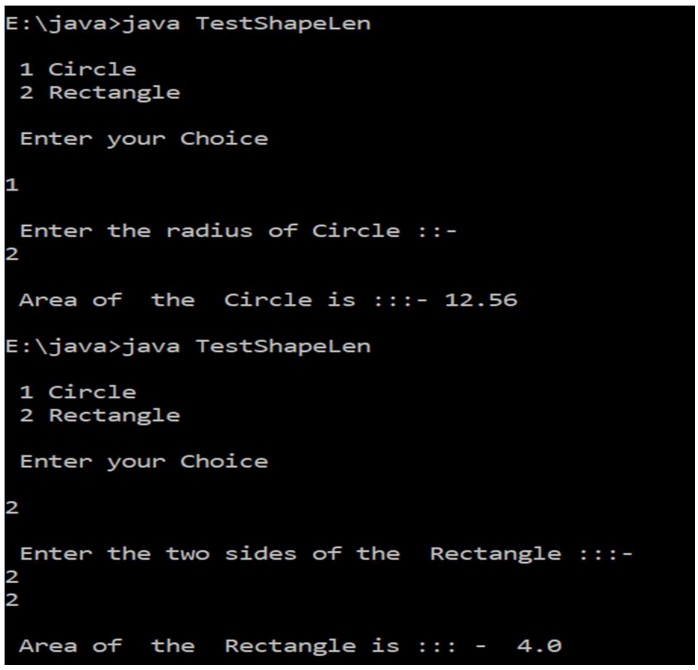
```
import java.util.Scanner;
abstract class Shape
{

   public double x;
        public float y;
```

```
        abstract  void   display();
        abstract  void  area();
}
class  Circle extends Shape
{
        private float radius;


        public void area()
        {

                System.out.println("\n Enter the radius of Circle ::-");
                Scanner sc= new Scanner(System.in);
                radius=sc.nextFloat();

                x=(3.14*radius*radius);
        }

        public void display()
        {

                System.out.println("\n Area of  the  Circle is :::- "+ x );
        }

}

class  Rectangle extends Shape
{

        private float  length;
        private  float width;

        public void area()
        {
                System.out.println("\n Enter the two sides of the  Rectangle :::- ");
                Scanner sc= new Scanner(System.in);
                width=sc.nextFloat();
                length= sc.nextFloat();
                y=(length*width);
        }

        public void display()
        {
                System.out.println("\n Area of  the  Rectangle is ::: -  "+ y);
        }

}
```

```
class TestShapeLen
{
	public static void main(String an[])
	{
		Shape ob;
		System.out.println(" \n 1 Circle \n 2 Rectangle");
		System.out.println("\n Enter your Choice \n");
		Scanner sc = new Scanner(System.in);
		int ch=sc.nextInt();
		if(ch==1)
		{
			ob= new Circle();

			ob.area();
			ob.display();
		}

		else if(ch==2)
		{
			ob = new Rectangle();

			ob.area();
			ob.display();
		}
	}
}
```

OUTPUT:-




E:\java>java TestShapeLen
1 Circle
2 Rectangle
Enter your Choice
1
Enter the radius of Circle ::-
2
Area of the Circle is :::- 12.56
E:\java>java TestShapeLen
1 Circle
2 Rectangle
Enter your Choice
2
Enter the two sides of the Rectangle :::-
2
2
Area of the Rectangle is ::: - 4.0