

1. Write a program in Java to create a class for employee record with two constructor, first default constructor to initialize all variables, second constructor to accept the employee details and a method to display employee details.

```
class Employee
{
    int e_no;
    double e_sal;
    String e_name;

    Employee()           //Default Constructor
    {
        e_no=101;
        e_sal=2090.50;
        e_name="AAA";
    }

    Employee(int empno,double empsal,String empname)    //Parametrized Constructor
    {
        e_no=empno;
        e_sal=empsal;
        e_name=empname;
    }
    void show()
    {
        System.out.println("Employee No. : "+e_no);
        System.out.println("Employee Salary. : "+e_sal);
        System.out.println("Employee Name. : "+e_name);
    }
}
class Demo_Constructor
{
    public static void main(String args[])
    {
        Employee e1=new Employee();
        e1.show();
        Employee e2=new Employee(102,50050,"BBB");
        e2.show();
    }
}
```

Output

Employee No. : 101
Employee Salary. : 2090.5
Employee Name. : AAA
Employee No. : 102
Employee Salary. : 50050.0
Employee Name. : BBB
Press any key to continue . . .

2. Write a program in Java that has Boolean function returning a string to display whether it is a leap year or not.

```
import java.io.*;

class Leap
{
    public boolean Check(int y)
    {
        if (y%4==0)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}

class Leap_demo
{
    public static void main(String args[])throws IOException
    {
        int yr;
        System.out.println("Enter the year:-");
        InputStreamReader isr=new InputStreamReader(System.in);
```

```

BufferedReader br=new BufferedReader(isr);
yr=Integer.parseInt(br.readLine());

Leap obj1=new Leap();
if(obj1.Check(yr)==true)
{
    System.out.println("Year is leap year");
}
else
{
    System.out.println("Year is not leap year");
}
}
}
}

```

Output

```

Enter the year:-
2020
Year is leap year
Press any key to continue . . .
Enter the year:-
2021
Year is not leap year
Press any key to continue . . .

```

3. Write a program in Java that has overloaded methods. The first methods should have no arguments. The second method should accept one string argument and the third method should accept one string and one integer argument. The first method should display “Delhi is an Important City” twice. The second method should display “Bombay is a beautiful city” thrice and third method should display “Chennai is a peaceful city” four times.

```

class Ovrload
{
    String s1,s2;
    int i,i1;

    void show()
    {

```

```

        System.out.println("Delhi is an important city.");
    }

    void show(String s)
    {
        s1=s;
        for(i=0;i<=2;i++)
        {
            System.out.println(s1);
        }
    }
    void show(String s,int i)
    {
        s2=s;
        i1=0;
        while(i1<i)
        {
            System.out.println(s2);
            i1++;
        }
    }
}
class Ovrload_demo
{
    public static void main(String args[])
    {
        Ovrload obj1=new Ovrload();
        obj1.show();
        obj1.show();

        System.out.println("\n");
        obj1.show("Bombay is a beautiful city");

        System.out.println("\n");
        obj1.show("Chennai is a peaceful city",4);
    }
}

```

Output

Delhi is an important city.

Delhi is an important city.

Bombay is a beautiful city
Bombay is a beautiful city
Bombay is a beautiful city

Chennai is a peaceful city
Chennai is a peaceful city
Chennai is a peaceful city
Chennai is a peaceful city
Press any key to continue . . .

4. Write a program in Java that creates an abstract class called Shape. Create subclasses that calculate and displays area of rectangle and triangle.

```
abstract class Shape
{
    int l,b,h,ba;
    abstract int area();
}
class rectangle extends Shape
{
    rectangle(int x ,int y)
    {
        l=x;
        b=y;
    }
    int area()
    {
        return l*b;
    }
}
```

```
class triangle extends Shape
{
    triangle (int a,int b)
    {
        ba=a;
        h=b;
    }
    int area()
    {
        return(ba*h)/2;
    }
}
```

```

    }
}

class Shape_demo
{
    public static void main(String args[])
    {
        Shape s1;
        rectangle R=new rectangle(10,20);
        triangle T=new triangle(20,40);

        int z;
        s1=R;
        z=s1.area();
        System.out.println("\n\n Area of rectangle: "+z);

        s1=T;
        z=s1.area();
        System.out.println("\n\n Area of Triangle: "+z);
    }
}

```

Output

Area of rectangle: 200

Area of Triangle: 400

Press any key to continue . . .

5. Write a program in Java to accept values for multithread.

```

class A extends Thread
{
    public void run()
    {
        for(int i=1;i<=5;i++)
        {
            System.out.println("from thread A: i="+i);
        }
        System.out.println(" Exit from thread A");
    }
}

```

```

class B extends Thread
{
    public void run()
    {
        for(int j=1;j<=5;j++)
        {
            System.out.println("from thread B: j="+j);
        }
        System.out.println(" Exit from thread B");
    }
}

class C extends Thread
{
    public void run()
    {
        for(int k=1;k<=5;k++)
        {
            System.out.println("from thread C: k="+k);
        }
        System.out.println(" Exit from thread C");
    }
}

class Thread_demo
{
    public static void main(String args[])
    {
        {
            new A().start();
            new B().start();
            new C().start();
        }
    }
}

```

Output

```

from thread A: i=1
from thread A: i=2
from thread C: k=1
from thread C: k=2
from thread C: k=3
from thread C: k=4
from thread C: k=5

```

Exit from thread C
from thread B: j=1
from thread B: j=2
from thread B: j=3
from thread B: j=4
from thread B: j=5
Exit from thread B
from thread A: i=3
from thread A: i=4
from thread A: i=5
Exit from thread A
Press any key to continue . . .

6. Write a Program to demonstrate multiple inheritance through interface.

```
class person
{
    String name;
    int age;
    String address;
    void persondetails(String nm, int ag, String add)
    {
        name=nm;
        age=ag;
        address=add;
    }
    void displayperson()
    {
        System.out.println("Name:"+name);
        System.out.println("Age:"+age);
        System.out.println("Address:"+address);
    }
}
class Employee extends person
{
    int empid;
    int salary;
    void empdetails(int id, int sal)
    {
        empid=id;
        salary=sal;
    }
}
```



```

    }
    void displayemployee()
{
    System.out.println("Empid:"+empid);
    System.out.println("Salary:"+salary);
}
}
interface Bonus
{
    int bonus=1000;
    void compute();
}
class Faculty extends Employee implements Bonus
{
    int amount;
    public void compute()
    {
        System.out.println("Bonus:"+bonus);
        amount=salary+bonus;
    }
    void facultydetails()
    {
        displayperson();
        displayemployee();
        compute();
        System.out.println("The total amount is:"+amount);
    }
}
public class MultipleInheritance
{
    public static void main(String[] args)
    {
        Faculty obj=new Faculty();
        obj.persondetails("Nisha",23,"115,Greenfield    Apartment,Pratap    Nagar,Nagpur-
440018");
        obj.empdetails(001, 20000);
        obj.facultydetails();

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

```

```
obj.persondetails("Surbhi", 27, "Sai Appartment,Sai Nagar,Nagpur-440023");

obj.empdetails(002,30000);
obj.facultydetails();
}
}
```

Output

Name:Nisha

Age:23

Address:115,Greenfield Apartment,Pratap Nagar,Nagpur-440018

Empid:1

Salary:20000

Bonus:1000

The total amount is:21000

Name:Surbhi

Age:27

Address:Sai Appartment,Sai Nagar,Nagpur-440023

Empid:2

Salary:30000

Bonus:1000

The total amount is:31000

Press any key to continue . . .

7. Write a Program to illustrate exception using multiple catch statement.

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

```
class TestMyException
{
    public static void main(String args[])
    {
        int x=5 , y=1000;
        try
        {
```

```

float z=(float)x/(float)y;
    if(z<0.01)
    {
        throw new MyException("number is too small");
    }
}
catch(MyException e)
{
    System.out.println("caught MyException");
    System.out.println(e.getMessage());
    System.out.println(e);
}
finally
{
    System.out.println("i am always here");
}
}
}

```

Output

```

caught MyException
number is too small
MyException: number is too small
i am always here
Press any key to continue . . .

```

8. Write a Java Program to create Lamp using applet.

```

import java.awt.*;

public class Lamp extends java.applet.Applet
{
    public void paint(Graphics g)
    {
        // the lamp platform
        g.fillRect(0,250,290,290);

        // the base of the lamp
        g.drawLine(125,250,125,160);
        g.drawLine(175,250,175,160);

        // the lamp shade, top and bottom edges
    }
}

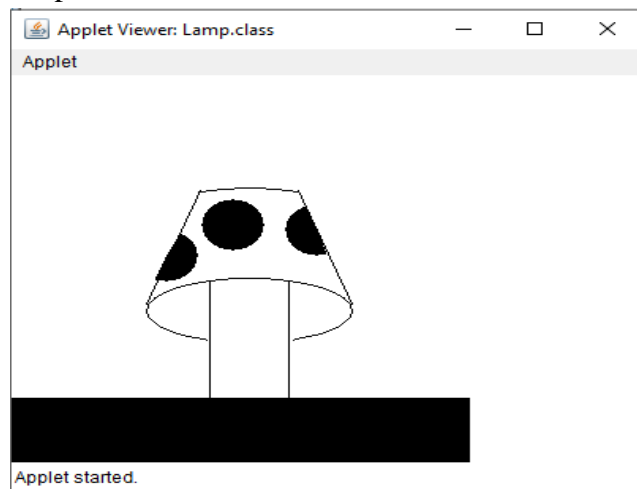
```

```
g.drawArc(85,157,130,50,-65,312);
g.drawArc(85,87,130,50,62,58);
```

```
// lamp shade, sides
g.drawLine(85,177,119,89);
g.drawLine(215,177,181,89);
```

```
// dots on the shade
g.fillArc(78,120,40,40,63,-174);
g.fillOval(120,96,40,40);
g.fillArc(173,100,40,40,110,180);
}
}
```

Output



9. Write a Program to create an application which shows the menu.

```
import java.awt.*;

class MyMenu extends Frame
{
    MenuBar bar;
    Menu menu1,menu2,menu3;
    MenuItem menuItem1,menuItem2,menuItem3,menuItem4;
    MyMenu(String s)
    {
        super(s);
        setSize(400,400);
    }
}
```

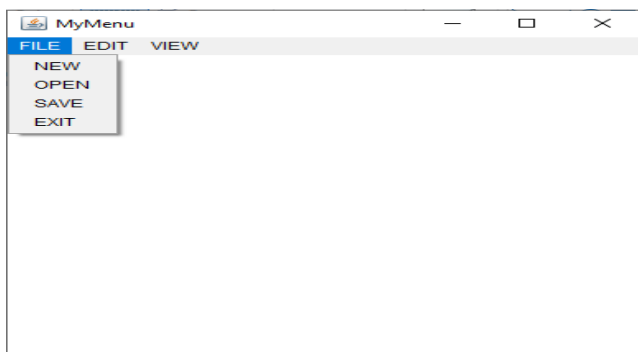
```

        bar = new MenuBar();
        menu1 = new Menu("FILE");
        menu2 = new Menu("EDIT");
        menu3 = new Menu("VIEW");
        menuItem1 = new MenuItem("NEW");
        menuItem2 = new MenuItem("OPEN");
        menuItem3 = new MenuItem("SAVE");
        menuItem4 = new MenuItem("EXIT");
        menu1.add(menuItem1);
        menu1.add(menuItem2);
        menu1.add(menuItem3);
        menu1.add(menuItem4);
        bar.add(menu1);
        bar.add(menu2);
        bar.add(menu3);
        setMenuBar(bar);
    }

    public static void main(String args[])
    {
        MyMenu m = new MyMenu("MyMenu");
        m.setVisible(true);
    }
}

```

Output



10. Write a Java Program for creation of input dialog box.

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

class ActionInputDialog extends JApplet implements ActionListener
{
    JFrame jf;
    JButton b1;
    JLabel l1;
    ActionInputDialog()
    {
        jf=new JFrame("Demo of an Input dialog box");
        l1=new JLabel();
        jf.setLayout(new FlowLayout());
        jf.setDefaultCloseOperation(jf.EXIT_ON_CLOSE);
        b1=new JButton("Name");
        b1.addActionListener(this);
        jf.setSize(350,200);
        jf.add(b1);
        jf.add(l1);
        jf.setVisible(true);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource() == b1)
        {
            int i = JOptionPane.QUESTION_MESSAGE;
            String pstr = JOptionPane.showInputDialog(jf,"What is your name?", "Input Dialog
Box",i);
            l1.setText("Your name is:"+pstr);
        }
    }
}
```

```
    }  
}  
public static void main(String str[])  
{  
    ActionInputDialog aid=new ActionInputDialog();  
}  
}
```

Output

