NITTE EDUCATION TRUST

NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

An autonomous institution with A Grade by NAAC UGC | Approved by UGC/ AICTE/ Govt. Of Karnataka | Bangalore

As a component of internal assessment method

Assignment Submitted to the Department of ISE

By

Name: Harsh Nautiyal

USN: 1NT17IS067

CLASS: 6th-A

Subject: Java Lab

Maximum Marks	
Marks Awarded	

Signature of the Faculty

1. Develop a small java application, which accepts employee id from the command prompt and displays the details using arrays.

```
import java.util.Date;
import java.util.Stack;
public class Employee1
{
	public static void main(String args[])
	{
		System.out.println("Enter Valid Employee ID : \n");
	//enter an id from the specified list
		int[] EmpId={1001,1002,1003,1004,1005,1006,1007};
	//valid array of IDs
		String[] EmpName={"Abc","Opqr","Ghi","Wxyz","Jklmn","Stuv","Def"};
	//array of names according to the IDs respectively
		String[]
JoinDate={"01/04/2009","23/08/2012","12/11/2008","29/01/2013","16/07/2005","01/01/200
0","12/06/2006"};  //array of joining dates according to the IDs respectively.
		char[] DesigCode={'e','c','k','r','m','e','c'};  //array of
designation codes according to the IDs respectively.
		String[]  Department={"R&D","PM","Acct","Front
Desk","Engg","Manufacturing","PM"};  //array of department of the employees according the
IDs respectively
		double[] Basic={20000,30000,10000,12000,50000,23000,29000};
	//array of basic salaries of employees.
		double[] HRA={8000,12000,8000,6000,20000,9000,12000};
	//array of hra of employees
		double[] IT={3000,9000,1000,2000,20000,4400,10000};
		//array of income tax of employees
		char[] DesignationCode={'e','c','k','r','m'};
		String[]
Designation={"Engineer","Consultant","Clerk","Receptionist","Manager"};//array  of
designations
		double[] DA={20000,32000,12000,15000,40000};
		//array of da of employees
		int flag=0;
		int id=Integer.parseInt(args[0]);

		for(int i=0;i<EmpId.length;i++)
		{
			if(EmpId[i]==id)
			{
				flag=1;
				System.out.println("Emp Id.	Emp Name	Department
	Designation	DA");		//printing employee details of specified employee id
				System.out.print(EmpId[i]+"	"+EmpName[i]+"
	"+Department[i]);
```

```
                        for(int j=0;j<DesignationCode.length;j++)
                        {
                                if(DesigCode[i]==DesignationCode[j])
                                {       System.out.print("          "+Designation[j]+"   ");
                                        double sum=Basic[i]+HRA[i]+DA[j]-IT[i];
                        //calculating sum
                                        System.out.print(sum);
                                }
                        }
                }
        }
        if(flag==0)
                System.out.println("There is no employee with EmpId : " +id);
    //prints this when an invalid employee ID is returned
    }
}
```

OUTPUT:

```
C:\>java E:\Downloads\Employee1.java 1001
Enter Valid Employee ID :

Emp Id.  Emp Name       Department      Designation        DA
1001      Abc           R&D             Engineer        45000.0
C:\>java E:\Downloads\Employee1.java 1005
Enter Valid Employee ID :

Emp Id.  Emp Name       Department      Designation        DA
1005      Jklmn                 Engg            Manager         90000.0
C:\>java E:\Downloads\Employee1.java 1007
Enter Valid Employee ID :

Emp Id.  Emp Name       Department      Designation        DA
1007      Def           PM              Consultant      63000.0
C:\>java E:\Downloads\Employee1.java 1020
Enter Valid Employee ID :

There is no employee with EmpId : 1020
```

2. Develop a small java application, which uses concepts of Multithreading

```
package javalab;
import java.util.Date;
import java.util.*;
public class MultiThreading implements Runnable//thread creation by implementing the Runnable Interface
{
	Thread t;
	static int[] a=new int[51];//creates array a
	static int sum=0;

	MultiThreading(String name)
	{
		t=new Thread(this, name);//creates new thread t
		System.out.println("childthread:"+t);
		t.start();//starts thread t
	}

	public void run()
	{
		System.out.println(Thread.currentThread().getName());//prints the current running thread
		if(Thread.currentThread().getName().compareTo("one")==0)
		{
			for(int i=0;i<10;i++)
			{
				sum=sum+a[i];
				try
				{
					Thread.sleep(1000);// Let the thread sleep for a while.
				}
				catch (InterruptedException e)//exception thrown when thread is interrupted
				{
					e.printStackTrace();
			}
			System.out.println("Sum of 1-10 : " +sum);
		}
			//total=total+sum;
	}
		else if(Thread.currentThread().getName().compareTo("two")==0)
		{
			for(int j=10;j<20;j++)
			{
				sum=sum+a[j];
				try
```

```
                        {
                                Thread.sleep(1000);// Let the thread sleep for a while.
                        }
                        catch (InterruptedException e)//exception thrown when thread is
interrupted
                        {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        }
                        System.out.println("Sum of 10-20 : " +sum);
                }
                //total=total+sum;
        }

        else if(Thread.currentThread().getName().compareTo("three")==0)
        {
                for(int k=20;k<30;k++)
                {
                        sum=sum+a[k];
                        try
                        {
                                Thread.sleep(1000);// Let the thread sleep for a while.
                        }
                        catch (InterruptedException e)//exception thrown when thread is
interrupted
                        {
                                e.printStackTrace();
                        }
                        System.out.println("Sum of 20-30 : " +sum);
                }
                //total=total+sum;
        }

        else if(Thread.currentThread().getName().compareTo("four")==0)
        {
                for(int l=30;l<40;l++)
                {
                        sum=sum+a[l];
                        try{
                                Thread.sleep(1000);// Let the thread sleep for a while.
                        }catch (InterruptedException e)//exception thrown when thread is
interrupted
                        {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        }
                        System.out.println("Sum of 30-40 : " +sum);
                }
```

```
                //total=total+sum;
        }

        else if(Thread.currentThread().getName().compareTo("five")==0)
        {
                for(int m=40;m<50;m++)
                {
                        sum=sum+a[m];
                        try
                        {
                                Thread.sleep(1000);// Let the thread sleep for a while.
                        }
                        catch (InterruptedException e)//exception thrown when thread is
interrupted
                        {
                                e.printStackTrace();
                        }
                        System.out.println("Sum of 40-50 : " +sum);
                        //total=total+sum;
                }
                //System.out.println("Total sum is : " +total);
        }
    }

    public static void main(String[] args)
    {
                for(int x=0;x<51;x++)
                {
                        a[x]=x+1;
                }

            System.out.println(Thread.currentThread().getName());
            MultiThreading ob1=new MultiThreading("one");
            MultiThreading ob2=new MultiThreading("two");
            MultiThreading ob3=new MultiThreading("three");
            MultiThreading ob4=new MultiThreading("four");
            MultiThreading ob5=new MultiThreading("five");

            Date start=new Date();

            System.out.println("First Thread is Alive? : " +ob1.t.isAlive());
            System.out.println("Second Thread is Alive? : " +ob2.t.isAlive());
            System.out.println("Third Thread is Alive? : " +ob3.t.isAlive());
            System.out.println("Fourth Thread is Alive? : " +ob4.t.isAlive());
            System.out.println("Fivth Thread is Alive? : " +ob5.t.isAlive());


            try
```

```
		{
			System.out.println("waiting for Threads to complete");
			ob1.t.join();
			ob2.t.join();
			ob3.t.join();
			ob4.t.join();
			ob5.t.join();
		}
		catch (InterruptedException e)
		{
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		System.out.println("Total sum is : " +sum);

		System.out.println("First Thread is Alive? :"+ob1.t.isAlive());
		System.out.println("Second Thread is Alive? :"+ob2.t.isAlive());
		System.out.println("Third Thread is Alive? :"+ob3.t.isAlive());
		System.out.println("Fourth Thread is Alive? :"+ob4.t.isAlive());
		System.out.println("Fivth Thread is Alive? :"+ob5.t.isAlive());

		System.out.println("Main thread is interupted ");

		Date end=new Date();
		long difference=end.getTime()-start.getTime();//time taken for execution
		System.out.println("Whole process took "+difference/1000 +" " +"seconds");
		System.out.println("Main thread is exiting");
	}
}
```

OUTPUT:

```
main
childthread:Thread[one,5,main]
childthread:Thread[two,5,main]
one
childthread:Thread[three,5,main]
two
three
childthread:Thread[four,5,main]
childthread:Thread[five,5,main]
four
five
First Thread is Alive? : true
Second Thread is Alive? : true
Third Thread is Alive? : true
Fourth Thread is Alive? : true
Fivth Thread is Alive? : true
waiting for Threads to complete
Sum of 1-10 : 105
Sum of 10-20 : 105
Sum of 20-30 : 105
Sum of 40-50 : 107
Sum of 30-40 : 107
Sum of 1-10 : 215
Sum of 20-30 : 218
Sum of 40-50 : 218
Sum of 10-20 : 218
Sum of 30-40 : 297
Sum of 40-50 : 330
Sum of 20-30 : 330
Sum of 10-20 : 330
Sum of 1-10 : 330
Sum of 30-40 : 416
Sum of 40-50 : 450
Sum of 10-20 : 495
Sum of 20-30 : 495
Sum of 1-10 : 535
Sum of 30-40 : 535
Sum of 40-50 : 575
Sum of 10-20 : 621
Sum of 20-30 : 621
Sum of 1-10 : 663
Sum of 30-40 : 669
Sum of 40-50 : 705
Sum of 10-20 : 752
Sum of 20-30 : 752
Sum of 1-10 : 796
Sum of 30-40 : 803
Sum of 40-50 : 840
Sum of 20-30 : 888
Sum of 30-40 : 888
Sum of 1-10 : 888
Sum of 10-20 : 888
Sum of 40-50 : 980
```

```
Select Command Prompt
Sum of 10-20 : 218
Sum of 30-40 : 297
Sum of 40-50 : 330
Sum of 20-30 : 330
Sum of 10-20 : 330
Sum of 1-10 : 330
Sum of 30-40 : 416
Sum of 40-50 : 450
Sum of 10-20 : 495
Sum of 20-30 : 495
Sum of 1-10 : 535
Sum of 30-40 : 535
Sum of 40-50 : 575
Sum of 10-20 : 621
Sum of 20-30 : 621
Sum of 1-10 : 663
Sum of 30-40 : 669
Sum of 40-50 : 705
Sum of 10-20 : 752
Sum of 20-30 : 752
Sum of 1-10 : 796
Sum of 30-40 : 803
Sum of 40-50 : 840
Sum of 20-30 : 888
Sum of 30-40 : 888
Sum of 1-10 : 888
Sum of 10-20 : 888
Sum of 40-50 : 980
Sum of 20-30 : 1029
Sum of 30-40 : 1058
Sum of 1-10 : 1097
Sum of 10-20 : 1106
Sum of 40-50 : 1125
Sum of 20-30 : 1175
Sum of 30-40 : 1205
Sum of 1-10 : 1245
Sum of 10-20 : 1245
Sum of 40-50 : 1275
Sum of 20-30 : 1275
Sum of 30-40 : 1275
Sum of 1-10 : 1275
Sum of 10-20 : 1275
Total sum is : 1275
First Thread is Alive? :false
Second Thread is Alive? :false
Third Thread is Alive? :false
Fourth Thread is Alive? :false
Fivth Thread is Alive? :false
Main thread is interupted
Whole process took 10 seconds
Main thread is exiting

C:\Users\imhri>
```

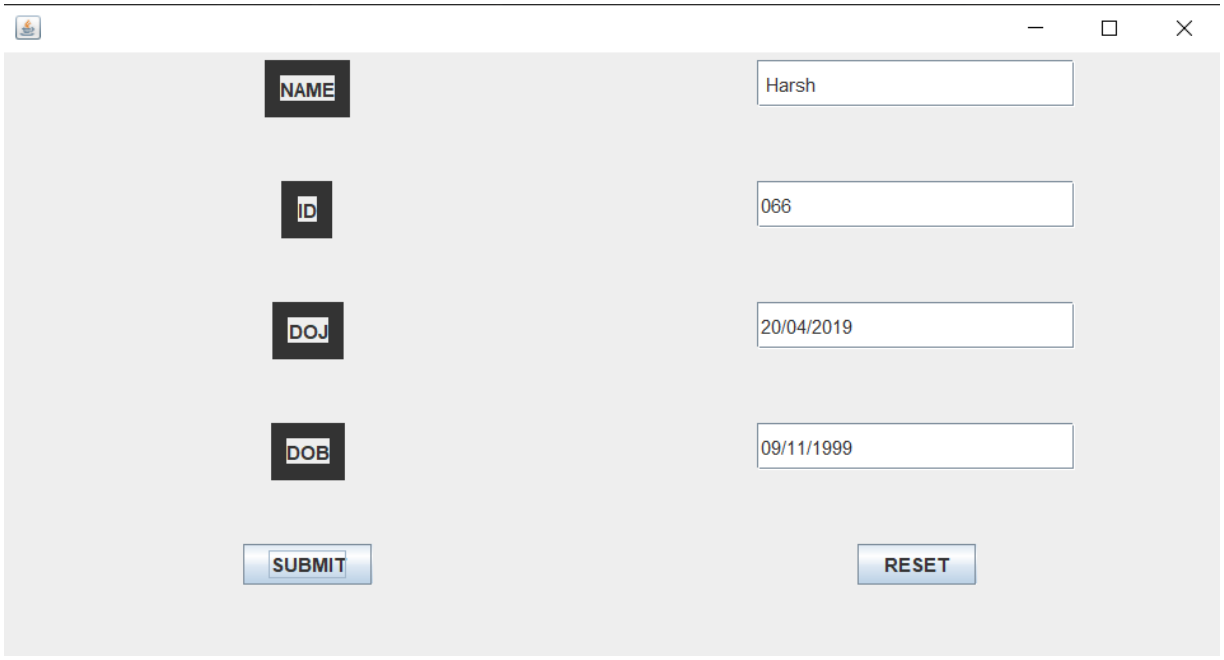
3. Design and Implement GUI for managing Employee Details using concepts of Files.

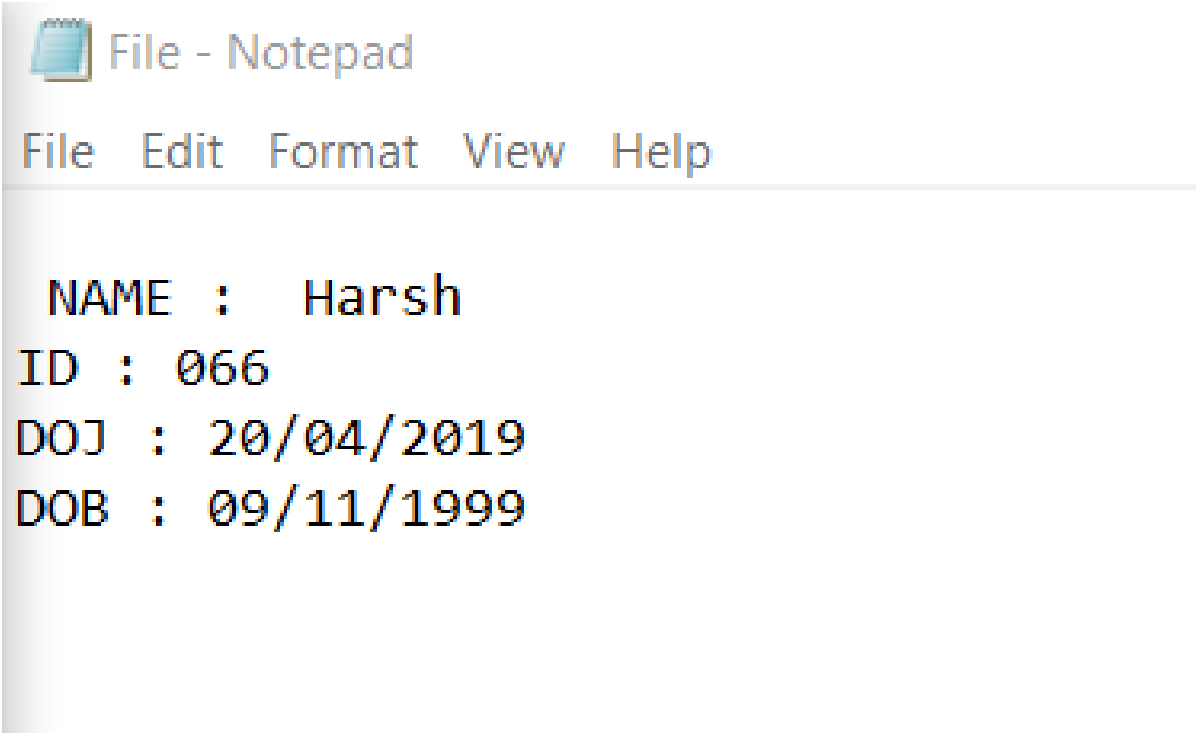
```
package javalab;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.*;

public class EmployeeFile
{
	private static Color black;

	public static void main(String[] args)
	{
		JFrame frameobj = new JFrame();  //creating frame
		frameobj.setSize(500, 500);  //declaring frame size

		GridLayout g1=new GridLayout(5,2);  // layout of the frame
		frameobj.setLayout(g1);                    //layout is set to the frame created

		JPanel p1=new JPanel();  //creating panels
		JPanel p2=new JPanel();
		JPanel p3=new JPanel();
		JPanel p4=new JPanel();
		JPanel p5=new JPanel();
		JPanel p6=new JPanel();
		JPanel p7=new JPanel();
		JPanel p8=new JPanel();
		JPanel p9=new JPanel();
		JPanel p10=new JPanel();

		JLabel l1=new JLabel("NAME");  //creating labels
		JLabel l2=new JLabel("ID");
		JLabel l3=new JLabel("DOJ");
		JLabel l4=new JLabel("DOB");

		JTextField f1=new JTextField();   //create object for text field
		JTextField f2=new JTextField();
		JTextField f3=new JTextField();
		JTextField f4=new JTextField();
```

```
f1.setPreferredSize(new Dimension(200,30)); //size of text field
f2.setPreferredSize(new Dimension(200,30));
f3.setPreferredSize(new Dimension(200,30));
f4.setPreferredSize(new Dimension(200,30));

JButton b1=new JButton("SUBMIT");//create submit button
JButton b2=new JButton("RESET");//create reset button

b1.addActionListener(new ActionListener()//is notified whenever you click on the button or menu item
{

	@Override
	public void actionPerformed(ActionEvent e)//is invoked automatically whenever you click on the registered component
	{

		File fileobj=new File("E:\\Downloads\\File.txt");//file path
		try
		{
			FileWriter fw=new FileWriter(fileobj.getAbsoluteFile(),true);
			System.out.println("\n NAME : " +f1.getText() +"\n" +"ID : " +f2.getText() +"\n" +"DOJ : " +f3.getText() +"\n" +"DOB : "+f4.getText() +"\n");//prints details
			fw.write("\n NAME : " +f1.getText() +"\n" +"ID : " +f2.getText() +"\n" +"DOJ : " +f3.getText() +"\n" +"DOB : "+f4.getText() +"\n");//writes details in file
			fw.close();
		}
		catch (IOException e1)
		{
			e1.printStackTrace();
		}
	}
});

b2.addActionListener(new ActionListener()
{
	@Override
	public void actionPerformed(ActionEvent e)
	{
		f1.setText(" ");
		f2.setText(null);
		f3.setText(null);
		f4.setText(null);
	}
```

```
        });

        p1.add(l1);   //add labels to panels where labels=name,id,doj,dob
        p3.add(l2);
        p5.add(l3);
        p7.add(l4);

        p2.add(f1);   //add text field to panels where text field is user defined
        p4.add(f2);
        p6.add(f3);
        p8.add(f4);

        p9.add(b1);    //add buttons to panel
        p10.add(b2);

        //l1.setBorder(BorderFactory.createLineBorder(Color.black));
        l1.setBorder(BorderFactory.createLineBorder(black,10));
        l2.setBorder(BorderFactory.createLineBorder(black,10));
        l3.setBorder(BorderFactory.createLineBorder(black,10));
        l4.setBorder(BorderFactory.createLineBorder(black,10));

        frameobj.add(p1);  //add panels to frames
        frameobj.add(p2);
        frameobj.add(p3);
        frameobj.add(p4);

        frameobj.add(p5);
        frameobj.add(p6);
        frameobj.add(p7);
        frameobj.add(p8);

        frameobj.add(p9);
        frameobj.add(p10);

        frameobj.setVisible(true);//shows the window
    }
}
```

OUTPUT:


NAME
Harsh
ID
066
DOJ
20/04/2019
DOB
09/11/1999
SUBMIT
RESET


File - Notepad
File Edit Format View Help
NAME : Harsh
ID : 066
DOJ : 20/04/2019
DOB : 09/11/1999

4.Design and implement a simple inventory central system for a small video rental store using constructors and Object List.

Video.java

```
package javalab;
public class Video
{
   String mName;//movie name
   boolean status;//status of return
   double rating;//movie rating
         public Video(String mName, boolean status, double rating)//constructor
         {
                  super();
                  this.mName = mName;//set movie name
                  this.status = status;//set return status
                  this.rating = rating;//set movie rating
         }
        public String getmName()
        {
                  return mName;//returns movie name
        }
        public void setmName(String mName)
        {
                  this.mName = mName;//sets movie name
        }
        public boolean isStatus()
        {
                  return status;//returns status
        }
        public void setStatus(boolean status)
        {
                  this.status = status;//sets status
        }
        public double getRating()
        {
                  return rating;//returns movie rating
        }
        public double setRating(double rating)
        {
                  return this.rating = rating;//sets movie rating
        }
}
```

VideoMethods.java

```
package javalab;
import java.util.List;
import java.util.Scanner;
import java.util.ArrayList;

public class VideoMethods
{
        List<Video> MovieList = new ArrayList<Video>();//create list
 public void AddMovies()//method to add movies into the list
 {
          Scanner in =new Scanner(System.in);
          System.out.print("Enter the name of the movie:");
          String mName=in.nextLine();
          System.out.print("Enter the status of the movie(True/False):");
          boolean status=in.nextBoolean();
          System.out.print("Enter the ratings for the movie(0-5):");
          double rating=in.nextDouble();
          Video v= new Video(mName, status, rating);//calls constructor
          MovieList.add(v);//inserts elements into list
            System.out.println("Library Initialized");
 }
          public void DisplayAll()//method to display list
          {
                  if(MovieList.isEmpty())//checks if list is empty
                  {
                          System.out.println("No movies in the library");
                  }
                  for(Video m : MovieList)//Iterates through whole loop
                  {
                          System.out.println("Movie   :   "   +m.getmName()+"      "+"Status   :
"+m.isStatus()+"  "+"Rating "+m.getRating());
                  }
          }

     boolean RentOut(String name)//method to rent movies
    {
          for(Video m :MovieList)
          {
                  if(m.getmName().equalsIgnoreCase(name))//checks if movie name is in the list
                  {
                          if(m.isStatus())
                          {
                                  m.setStatus(false);
                                  return true;
                          }
```

```
                }
                return false;
        }
                return false;
    }

    public void CollectIn(String name,double rat)//method to collect back rented movies
    {
        boolean flag=false;
        for(Video m :MovieList)
        {
                if(m.getmName().equalsIgnoreCase(name))//checks if movie name is in the list
                {
                                m.setStatus(true);
                                flag=true;
                        Math.round(m.setRating((m.getRating() + rat)/2));
                }
        }
        if(!flag)//movie name is not rented
        {
                System.out.println("Requested Movie not rented out");
        }
    }
}
```

Videomain.java

```
package javalab;
import java.util.Scanner;
public class VideoMain
{
        public static void main(String args[])
        {
                VideoMethods mm = new VideoMethods();// new object created
                while(true)
                {
                                System.out.println("%%%%%%%%      VIDEO      LIBRARY
CENTER %%%%%%%%");
                                int n;//for choice
                                Scanner in = new Scanner(System.in);
                                System.out.println("1.ADD MOVIES");
                                System.out.println("2.DISPLAY MOVIES");
                                System.out.println("3.RENT OUT");
                                System.out.println("4.COLLECT BACK ");

                                System.out.println("PLEASE ENTER YOUR OPTION");
```

```
                        n = in.nextInt();
                        switch(n)
                        {
                                case 1:mm.AddMovies();
                                break;
                                case 2:mm.DisplayAll();
                                break;
                                case 3:System.out.print("Enter the movie you want to
rent.");
                                in.nextLine();
                                if(mm.RentOut(in.nextLine()))//calling method RentOut
                                {
                                        System.out.println("Rent out successfull");
                                }
                                else
                                {
                                        System.out.println("Sorry!! Not Available");
                                }
                                break;
                                case 4:System.out.println("Enter the name and the ratings
of the movie");
                                in.nextLine();
                                mm.CollectIn(in.nextLine(),in.nextDouble());//calling
method CollectIn
                                break;
                        }
                }
        }
}
```

OUTPUT:

```
%%%%%%%% VIDEO LIBRARY CENTER %%%%%%%%
1.ADD MOVIES
2.DISPLAY MOVIES
3.RENT OUT
4.COLLECT BACK
PLEASE ENTER YOUR OPTION
1
Enter the name of the movie:Avengers
Enter the status of the movie(True/False):true
Enter the ratings for the movie(0-5):5
Library Initialized
%%%%%%%% VIDEO LIBRARY CENTER %%%%%%%%
1.ADD MOVIES
2.DISPLAY MOVIES
3.RENT OUT
4.COLLECT BACK
PLEASE ENTER YOUR OPTION
2
Movie : Avengers  Status : true  Rating 5.0
%%%%%%%% VIDEO LIBRARY CENTER %%%%%%%%
1.ADD MOVIES
2.DISPLAY MOVIES
3.RENT OUT
4.COLLECT BACK
PLEASE ENTER YOUR OPTION
3
Enter the movie you want to rent.Avengers
Rent out successfull
%%%%%%%% VIDEO LIBRARY CENTER %%%%%%%%
1.ADD MOVIES
2.DISPLAY MOVIES
3.RENT OUT
4.COLLECT BACK
PLEASE ENTER YOUR OPTION
3
Enter the movie you want to rent.Avengers
Sorry!! Not Available
%%%%%%%% VIDEO LIBRARY CENTER %%%%%%%%
1.ADD MOVIES
2.DISPLAY MOVIES
3.RENT OUT
4.COLLECT BACK
PLEASE ENTER YOUR OPTION
4
Enter the name and the ratings of the movie
Avengers 5
```

5.Given the information about employees of an organization, develop a small java application, using JDBC.

DBConnection.java

```
package javalab;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection
{
	public Connection getDBconnection()
	{
		Connection conn=null;
		try
		{
			Class.forName("com.mysql.jdbc.Driver");//registering mysql drivers
			System.out.println("Registered successfully");
		}
		catch (ClassNotFoundException e)
		{
			e.printStackTrace();
		}
		try
		{

	conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/EMP","root","");//
connecting to database
			System.out.println("Connection successfull\n");
		}
		catch (SQLException e)
		{
			e.printStackTrace();
		}
		return conn;
	}
}
```

Employee.java

```
package javalab;
public class Employee
{
	String name;
	int age;
```

```
	String dept;
	double sal;
	public String getName()
	{
		return name;//returns employee name
	}
	public void setName(String name)
	{
		this.name = name;//sets employee name
	}
	public int getAge()
	{
		return age;//returns employee age
	}
	public void setAge(int age)
	{
		this.age = age;//sets employee age
	}
	public String getDept()
	{
		return dept;//returns employee department
	}
	public void setDept(String dept)
	{
		this.dept = dept;//sets employee department
	}
	public double getSal()
	{
		return sal;//returns employee salary
	}
	public void setSal(double sal)
	{
		this.sal = sal;//sets employee salary
	}
}
```

DAO.java

```
package javalab;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class DAO
```

```
{

	public int insertemp(Employee e)//method to insert employee details
	{
		int r=0;
		Connection conn=new DBConnection().getDBconnection();
		try
		{
			PreparedStatement pst=conn.prepareStatement("insert into employee values(?,?,?,?)");//accepts input parameters at runtime
			pst.setString(1, e.getName());//Sets the designated parameter to the given Java String value
			pst.setInt(2, e.getAge());//Sets the designated parameter to the given Java int value
			pst.setString(3, e.getDept());//Sets the designated parameter to the given Java String value
			pst.setDouble(4, e.getSal());//Sets the designated parameter to the given Java double value
			r=pst.executeUpdate();//Executes the SQL statement

		}
		catch (SQLException e1)
		{
			e1.printStackTrace();
		}
		return r;
	}

	public int deleteemp(String emp)//method to delete specific employee details
	{
		int s=0;
		Connection conn=new DBConnection().getDBconnection();
		try
		{
			PreparedStatement pst=conn.prepareStatement("delete from employee where name=?");//accepts parameters to delete details
			pst.setString(1, emp);//Sets the designated parameter to the given Java String value
			s=pst.executeUpdate();//Executes the SQL statement
		}
		catch (SQLException e1)
		{
			e1.printStackTrace();
		}
		return s;
	}

	public void displayname(String ename)//method to display specific employee details
```

```
	{
		Connection conn=new DBConnection().getDBconnection();
		try
		{
			PreparedStatement pst = conn.prepareStatement("select * from employee where name=?");//accepts parameters to display details
			pst.setString(1, ename);//Sets the designated parameter to the given Java String value
			ResultSet rs=pst.executeQuery();//refers to the row and column data contained in a ResultSet object.
			while(rs.next())
			{
				System.out.println("Name : " +rs.getString(1)+ "\t"+ "Age : " +rs.getInt(2)+ "\t"+ "Dept :" +rs.getString(3)+ "\t"+ "Salary :" +rs.getDouble(4));//displays details
			}
		}
		catch (SQLException e)
		{
			e.printStackTrace();
		}

	}

	public void displayall()////method to display all employee details
	{
		Connection conn=new DBConnection().getDBconnection();
		try
		{
			PreparedStatement pst=conn.prepareStatement("select * from employee");
			ResultSet rs=pst.executeQuery();//refers to the row and column data contained in a ResultSet object.
			while(rs.next())
			{
				System.out.println("Name : " +rs.getString(1)+ "\t"+ "Age : " +rs.getInt(2)+"\t"+ "Dept :" +rs.getString(3)+"\t" + "Salary :" +rs.getDouble(4));;
			}
		}
		catch (SQLException e1)
		{
			e1.printStackTrace();
		}
	}
}
```

FinalDBProgram.java

```
package javalab;
import java.util.Scanner;

public class FinalDBProgram
{
	public static void main(String[] args)
	{
		for( ; ; )
		{
			Scanner in=new Scanner(System.in);
			System.out.println("\n 1. Insert Emp \n 2. Delete Emp \n 3. Display Acc to Name \n 4. Display All \n 5. Exit \n");
			System.out.println("Enter your choice");
			int n=in.nextInt();

			Employee e=new Employee();//object of class Employee
			DAO d=new DAO();//object of class DAO

			switch(n)
			{
			case 1:
					System.out.println("Enter the Employee Name : ");
					e.setName(in.next());
					System.out.println("Enter the Age : ");
					e.setAge(in.nextInt());
					System.out.println("Enter the Dept : ");
					e.setDept(in.next());
					System.out.println("Enter the Salary : ");
					e.setSal(in.nextDouble());
					d.insertemp(e);
					System.out.println("Employee added successfully");
					break;
			case 2:
					System.out.println("Enter the employee name : ");
					String Newname=in.next();
					d.deleteemp(Newname);
					System.out.println("Employee deleeted successfully");
					break;
			case 3:System.out.println("Enter the Employee name to display its attributes : ");
					String ename=in.next();
					System.out.println("Employee details...");
					d.displayname(ename);
					break;
			case 4:System.out.println("Employee deatils are as follows...");
					d.displayall();
```

```
                    break;
                case 5:System.exit(0);
                    break;
                    default:
                        System.out.println("Please Choose Valid option \n");
                        break;
                }
            }
        }
}
```

OUTPUT:

```
mysql> create database project;
Query OK, 1 row affected (0.31 sec)

mysql> use project;
Database changed
mysql> create table employee(name varchar(20) primary key,
    -> age int,
    -> department varchar(20),
    -> salary decimal(20,1));
Query OK, 0 rows affected (1.31 sec)

mysql> insert into employee values('aayush',21,'ISE',50000.0),('harsh',20,'ISE',45000.0),('hrithik',20,'ISE',75000.0);
Query OK, 3 rows affected (0.25 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from employee;
+---------+------+------------+---------+
| name    | age  | department | salary  |
+---------+------+------------+---------+
| aayush  |   21 | ISE        | 50000.0 |
| harsh   |   20 | ISE        | 45000.0 |
| hrithik |   20 | ISE        | 75000.0 |
+---------+------+------------+---------+
3 rows in set (0.02 sec)
```

```
<terminated> FinalDBProgram (1) [Java Application]
1. Insert Emp
2. Delete Emp
3. Display Acc to Name
4. Display All
5. Exit

Enter your choice
4
Employee deatils are as follows...
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is automatically registered vi
Registered successfully
Connection successfull

Name : aayush   Age : 21       Dept :ISE      Salary :50000.0
Name : harsh    Age : 20       Dept :ISE      Salary :45000.0
Name : hrithik  Age : 20       Dept :ISE      Salary :75000.0
```

```
 1. Insert Emp
 2. Delete Emp
 3. Display Acc to Name
 4. Display All
 5. Exit

Enter your choice
1
Enter the Employee Name :
aishwarya
Enter the Age :
21
Enter the Dept :
ISE
Enter the Salary :
100000.0
Registered successfully
Connection successfull

Employee added successfully
```

```
mysql> select * from employee;
+-----------+------+------------+----------+
| name      | age  | department | salary   |
+-----------+------+------------+----------+
| aayush    |   21 | ISE        |  50000.0 |
| aishwarya |   21 | ISE        | 100000.0 |
| harsh     |   20 | ISE        |  45000.0 |
| hrithik   |   20 | ISE        |  75000.0 |
+-----------+------+------------+----------+
4 rows in set (0.00 sec)
```

```
 1. Insert Emp
 2. Delete Emp
 3. Display Acc to Name
 4. Display All
 5. Exit

Enter your choice
2
Enter the employee name :
aishwarya
Registered successfully
Connection successfull

Employee deleeted successfully

 1. Insert Emp
 2. Delete Emp
 3. Display Acc to Name
 4. Display All
 5. Exit

Enter your choice
3
Enter the Employee name to display its attributes :
aayush
Employee details...
Registered successfully
Connection successfull

Name : aayush   Age : 21        Dept :ISE       Salary :50000.0

 1. Insert Emp
 2. Delete Emp
 3. Display Acc to Name
 4. Display All
 5. Exit

Enter your choice
5
```

```
mysql> select * from employee;
+---------+------+------------+---------+
| name    | age  | department | salary  |
+---------+------+------------+---------+
| aayush  |   21 | ISE        | 50000.0 |
| harsh   |   20 | ISE        | 45000.0 |
| hrithik |   20 | ISE        | 75000.0 |
+---------+------+------------+---------+
3 rows in set (0.00 sec)
```