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

Assignment Submitted to the Department of ISE

By

Name: Aashna Sinha

USN: 1NT17IS003

Semester: VI 'A'

Subject: JAVA Lab(Part-B)

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.

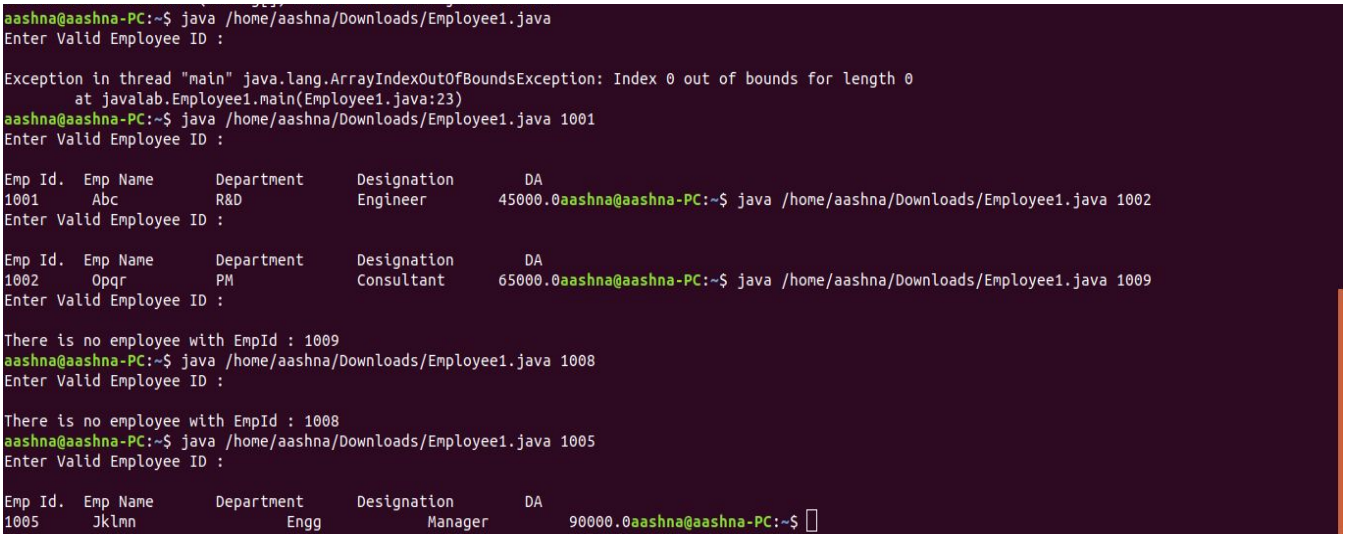
```
 package javalab;
import java.util.Date;
import java.util.Stack;


public class Employee1
{
	public static void main(String[] args)
	{
		System.out.println("Enter Valid Employee ID : \n");
		int[]
EmpId={1001,1002,1003,1004,1005,1006,1007};//initialize
employee id array
		String[]
EmpName={"Abc","Opqr","Ghi","Wxyz","Jklmn","Stuv","Def"}
;//initialize employee name array
		String[]
JoinDate={"01/04/2009","23/08/2012","12/11/2008","29/01/2013
","16/07/2005","01/01/2000","12/06/2006"};//initialize  employee
date of birth array
```

```
        char[] DesigCode={'e','c','k','r','m','e','c'};//initialize design code array
        String[] Department={"R&D","PM","Acct","Front Desk","Engg","Manufacturing","PM"};//initialize department array
        double[] Basic={20000,30000,10000,12000,50000,23000,29000};//initialize employee salary array
        double[] HRA={8000,12000,8000,6000,20000,9000,12000};//initialize hra array
        double[] IT={3000,9000,1000,2000,20000,4400,10000};//initialize it array
        char[] DesignationCode={'e','c','k','r','m'};//initialize designation code array
        String[] Designation={"Engineer","Consultant","Clerk","Receptionist","Manager"};//initialize designation array
        double[] DA={20000,32000,12000,15000,40000};//initialize da array
        int flag=0;//to check if id is found or not
        int id=Integer.parseInt(args[0]);//assign the id we have to find
```

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

if(DesigCode[i]==DesignationCode[j])//match desigcodes
                        {     System.out.print("
"+Designation[j]+" ");//print designation
                            double
sum=Basic[i]+HRA[i]+DA[j]-IT[i];//calculate salary
                            System.out.print(sum);//          print
salary
                        }
                    }
                }
            }
```

```
            if(flag==0)//invalid employee id
                System.out.println("There is no employee with EmpId : " +id);
        }
}
```

Output



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++)
                                                            {
```

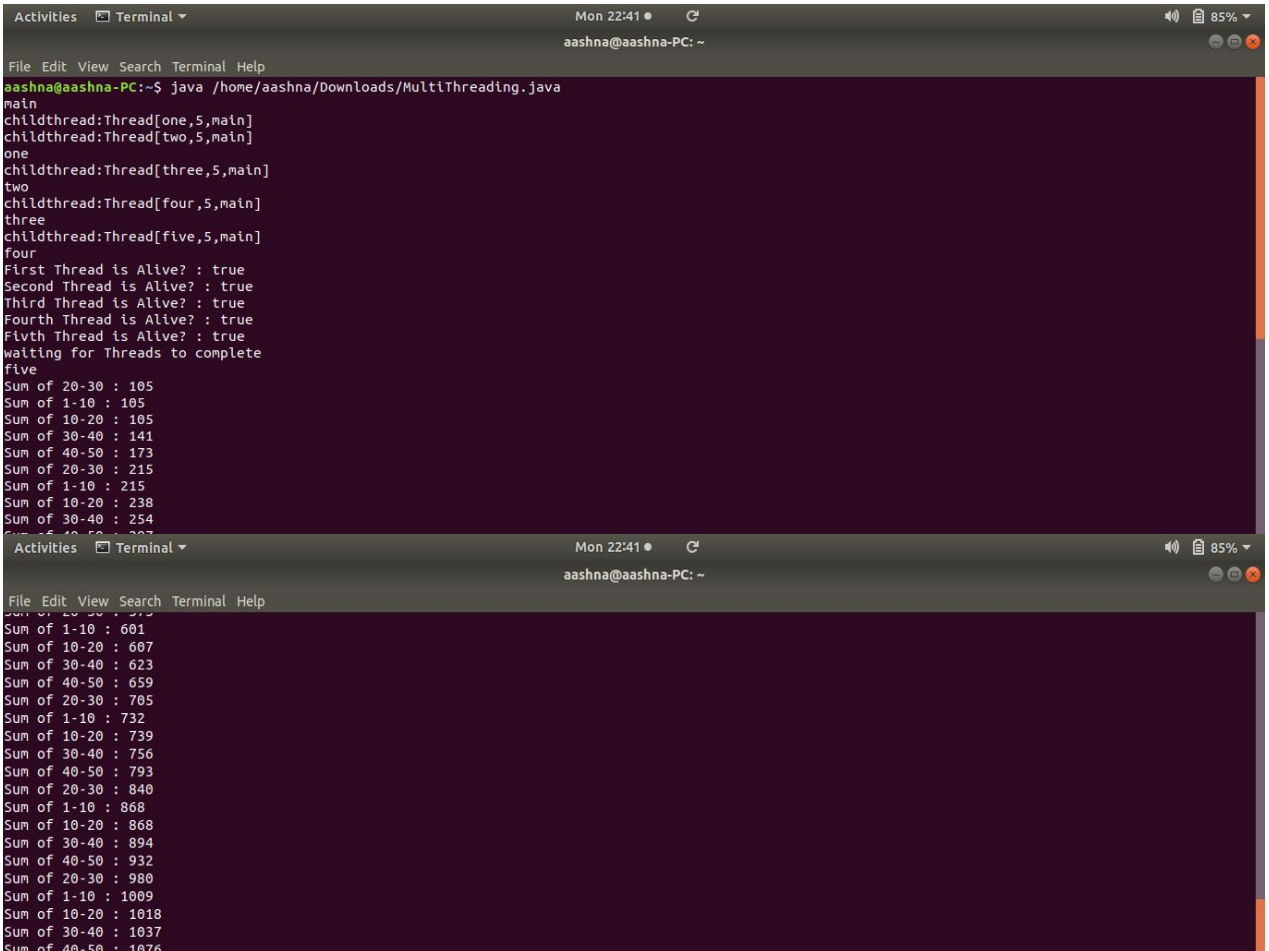
```
a[x]=x+1;
                                                    }

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



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
```

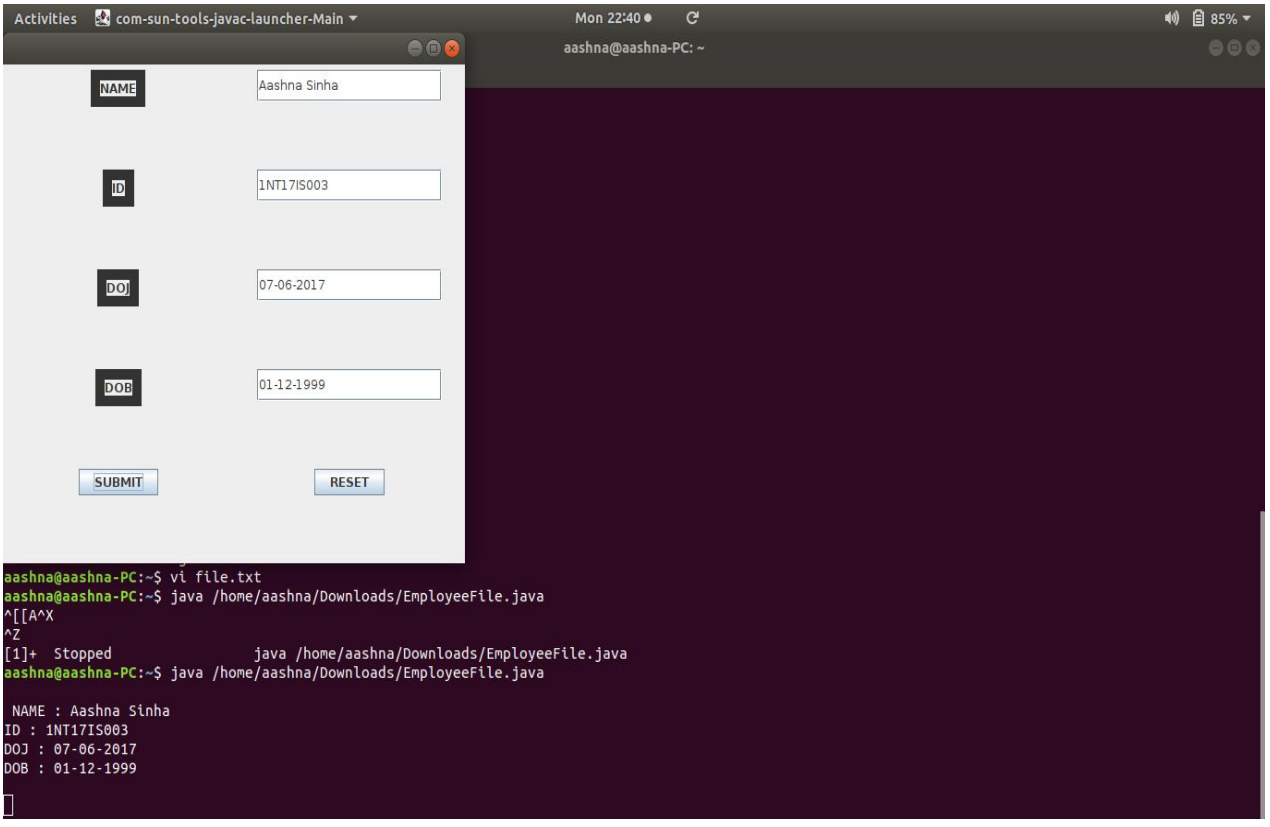
```
                                                    public
void actionPerformed(ActionEvent e)//is invoked automatically
whenever you click on the registered component
                                                    {

                                                        File
fileobj=new File("/home/aashna/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



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

Bean Class File : Video.java

```
public class Video
{
    String mName;
    boolean status;
    double rating;
    public Video(String mName, boolean status, double rating)
    {
       super();
       this.mName = mName;
       this.status = status;
       this.rating = rating;
    }
   public String getmName()
   {
      return mName;
   }
   public void setmName(String mName)
   {
      this.mName = mName;
   }
   public boolean isStatus()
   {
      return status;
```

```
    }
    public void setStatus(boolean status)
    {
        this.status = status;
    }
    public double getRating()
    {
        return rating;
    }
    public double setRating(double rating)
    {
        return this.rating = rating;
    }
}
```

Methods Class File : VideoMethods.java

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

public class VideoMethods
{
    List<Video> MovieList = new ArrayList<Video>();
  public void AddMovies()
  {
      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);
    MovieList.add(v);
      System.out.println("Library Initialized");
  }
    public void DisplayAll()
    {
       if(MovieList.isEmpty())
       {
          System.out.println("No movies in the library");
       }
       for(Video m : MovieList)
       {
          System.out.println("Movie : " +m.getmName()+"
"+"Status : "+m.isStatus()+"  "+"Rating "+m.getRating());
       }
    }

     boolean RentOut(String name)
    {
       for(Video m :MovieList)
       {
          if(m.getmName().equalsIgnoreCase(name))
          {
             if(m.isStatus())
```

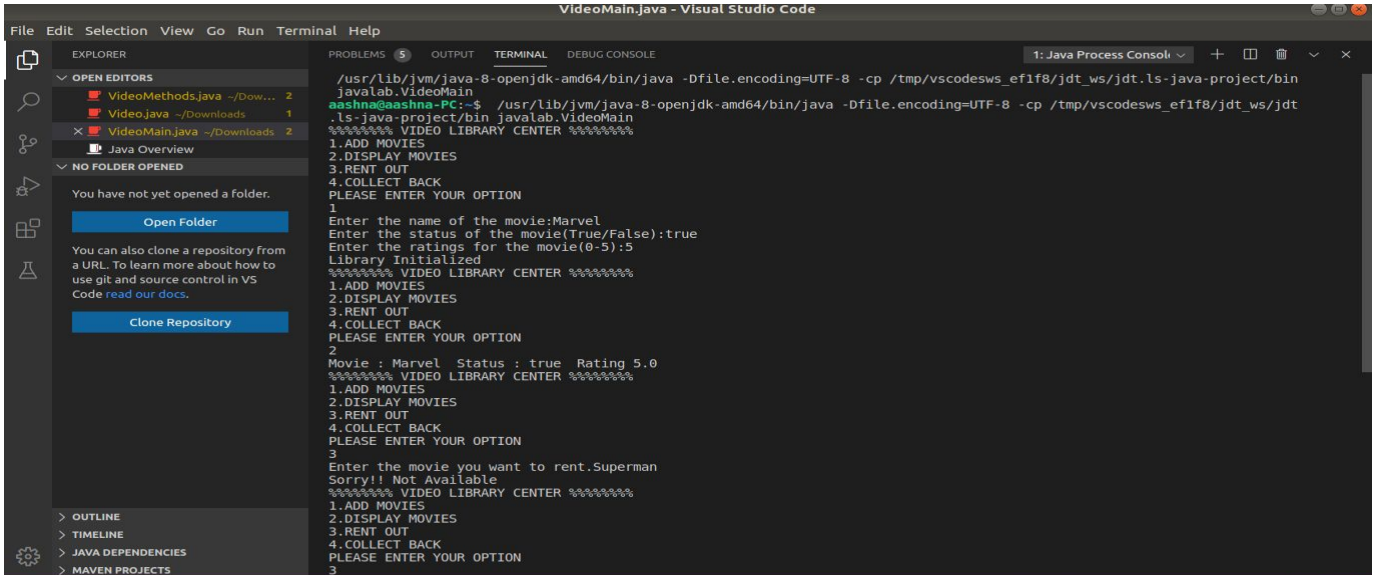
```
                {
                    m.setStatus(false);
                    return true;
                }
            }
            return false;
        }
      return false;
    }

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

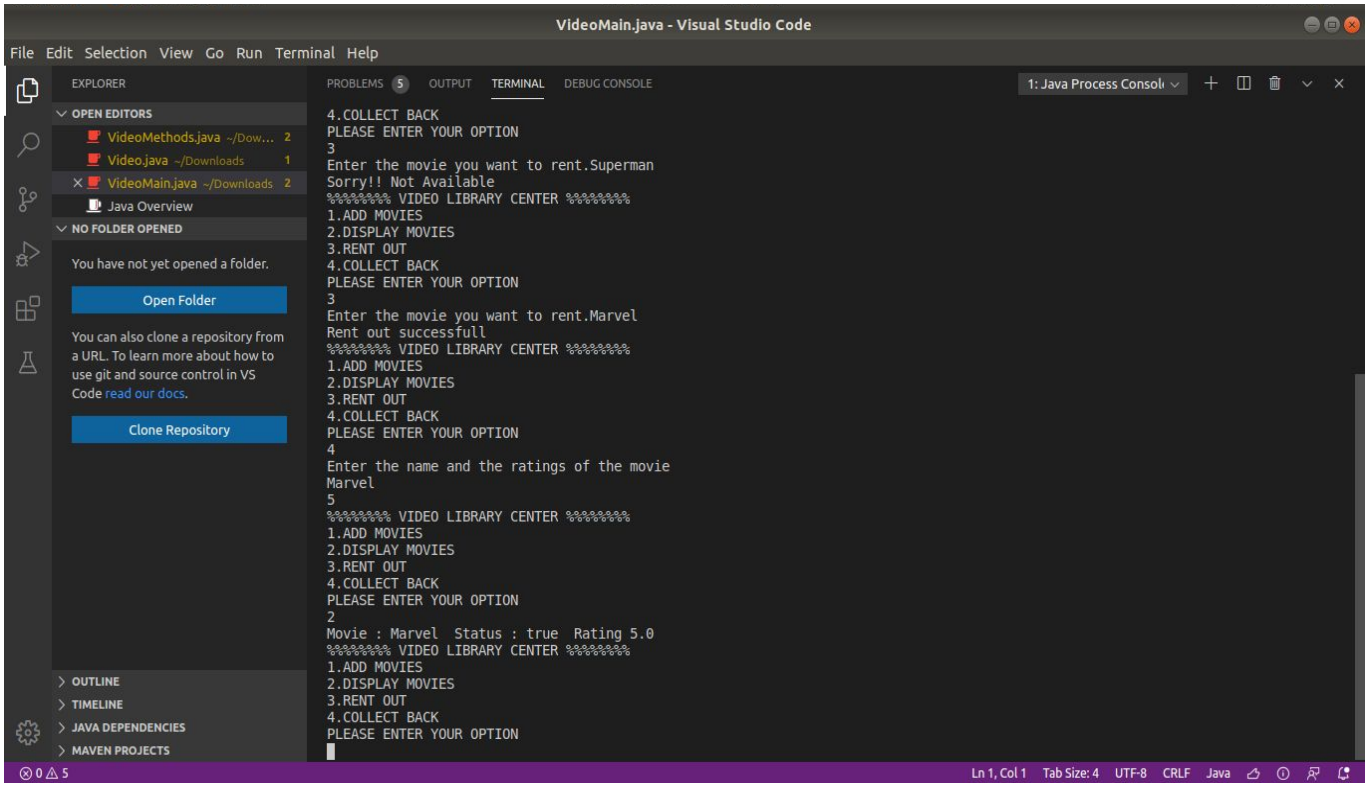
Main Class File: VideoMain.java

```
import java.util.Scanner;

public class VideoMain
{
   public static void main(String args[])
   {
      VideoMethods mm = new VideoMethods();
      while(true)
      {
            System.out.println("%%%%%%%% VIDEO
LIBRARY CENTER %%%%%%%%");
            int n;
            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()))
            {
               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());
            break;
         } }
   }
   }
```

Output





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

DBConnection.java

```
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");
            System.out.println("Registered successfully");
        }
        catch (ClassNotFoundException e)
        {
            e.printStackTrace();
        }
        try
        {

conn=DriverManager.getConnection("jdbc:mysql://localhost:330
6/EMP","root","");
            System.out.println("Connection successfull\n");
        }
        catch (SQLException e)
        {
            e.printStackTrace();
        }
        return conn;
    }
}
```

Employee.java

```
public class Employee
{
    String name;
```

```
int age;
String dept;
double sal;
public String getName()
{
   return name;
}
public void setName(String name)
{
   this.name = name;
}
public int getAge()
{
   return age;
}
public void setAge(int age)
{
   this.age = age;
}
public String getDept()
{
   return dept;
}
public void setDept(String dept)
{
   this.dept = dept;
}
public double getSal()
{
```

```
        return sal;
    }
    public void setSal(double sal)
    {
        this.sal = sal;
    }
}
```

DAO.java

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

public class DAO
{

    public int insertemp(Employee e)
    {
        int r=0;
        Connection conn=new DBConnection().getDBconnection();
        try
        {
            PreparedStatement pst=conn.prepareStatement("insert
into employee values(?,?,?,?)");
            pst.setString(1, e.getName());
            pst.setInt(2, e.getAge());
```

```
            pst.setString(3, e.getDept());
            pst.setDouble(4, e.getSal());
            r=pst.executeUpdate();
        }
        catch (SQLException e1)
        {
            e1.printStackTrace();
        }
        return r;
    }

    public int deleteemp(String emp)
    {
        int s=0;
        Connection conn=new DBConnection().getDBconnection();
        try
        {
            PreparedStatement pst=conn.prepareStatement("delete
from employee where name=?");
            pst.setString(1, emp);
            s=pst.executeUpdate();
        }
        catch (SQLException e1)
        {
            e1.printStackTrace();
        }
        return s;
    }
```

```
    public void displayname(String ename)
    {
       Connection conn=new DBConnection().getDBconnection();
       try
       {
            PreparedStatement pst = conn.prepareStatement("select
* from employee where name=?");
            pst.setString(1, ename);
            ResultSet rs=pst.executeQuery();
            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 e)
       {
            e.printStackTrace();
       }

    }

    public void displayall()
    {
       Connection conn=new DBConnection().getDBconnection();
       try
       {
```

```
            PreparedStatement pst=conn.prepareStatement("select *
from employee");
            ResultSet rs=pst.executeQuery();
            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

```
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();
        DAO d=new 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:

Create Database:

```
create database emp;
use emp;
create table employee
( Name varchar(20) primary key,
  Age int not null,
  Department varchar(20) not null,
  Salary int not null
```

);

Console:
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...

Registered successfully

Connection successfull

Name : abc Age : 25 Dept :ISE Salary :25000.0

Name : pqr Age : 30 Dept :ISE Salary :30000.0

1. Insert Emp

2. Delete Emp

3. Display Acc to Name

4. Display All

5. Exit

Enter your choice

1

Enter the Employee Name :

Def

Enter the Age :

28

Enter the Dept :

CSE

Enter the Salary :

26000

Registered successfully

Connection successfull

Employee added successfully

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...

Registered successfully

Connection successfull

Name : abc Age : 25 Dept :ISE Salary :25000.0

Name : pqr Age : 30 Dept :ISE Salary :30000.0

Name : Def Age : 28 Dept :CSE Salary :26000.0

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 :
Abc
Employee details...
Registered successfully
Connection successfull

Name : abc Age : 25 Dept :ISE Salary :25000.0

1. Insert Emp
2. Delete Emp
3. Display Acc to Name
4. Display All
5. Exit
Enter your choice
2
Enter the employee name :
Abc
Registered successfully
Connection successful

Employee deleted successfully

1. Insert Emp
2. Delete Emp
3. Display Acc to Name

4. Display All
5. Exit

Enter your choice
4
Employee details are as follows...
Registered successfully
Connection successfull

Name : pqr Age : 30 Dept :ISE Salary :30000.0
Name : Def Age : 28 Dept :CSE Salary :26000.0

View Changes.

select * from employee;

Name	Age	Department	Salary
abc	25	ISE	25000
pqr	30	ISE	30000
Def	28	CSE	26000