**Experiment -1.4**

**Student Name: Parikshit UID:** 19BCS4520

## Branch: CSE-IOT1 Section/Group: 1/A

**Semester:** 4

**Subject Name:** Project Based Learning in Java

**Subject Code:** CSP-296

## Date of Performance:

1. **Aim/Overview of the practical:**

### Create a menu based Java application with the following options: 1.Add an Employee

1. Display All 3.Exit

If option 1 is selected, the application should gather details of the employee like employee name, employee id, designation and salary and store it in a file. If option 2 is selected, the application should display all the employee details. If option 3 is selected the application should exit.

1. **Task to be done:**
   * Create a menu based Java application
   * Option 1 : creating or adding an employee details
   * Option 2 : writing employee details
   * Option 3 : reading employee details
   * Option 4 : displaying employee details
   * Option 5 : exit
2. **Apparatus(For applied/experimental sciences/materials based labs):** nil
3. **Algorithm/Flowchart (For programming based labs):**

### Step 1 :- Create a class named **Employee\_6098 ,** having all the required member variables and member functions.

* Step 2 :- Member variables declared here are – empID of type integer , name, designation , salary , contact , all of type String.
* Step 3 :- Call a constructor of the class, passing empID and name as its argument.
* Step 4 :- Make setter and getter methods of all the fields for providing them information and using as per the need.
* Step 5 :- Create a class named **EmpManager\_6098 ,** which will define the functionalities required.
* Step 6 :- Functionalities to be defined in the class are – adding an employee , writing details of employee , reading details of employee, and displaying the records.
* Step 7 :- Create a class named **Launcher\_6098 ,** which contains the main method to test our logic/program.
* Step 8 :- Create a main menu to ask the user for his choice among adding , writing , reading , or displaying the employee details.
* Step 9 :- Testing our program against all possible test cases.

1. **Theme/Interests definition( For creative domains):** nil
2. **Steps for experiment/practical:** Source Code :-

### Employee Class ::

**import** java.io.Serializable;

**public class** Employee\_6098 **implements** Serializable

{

**int** empID\_6098;

String name\_6098, designation\_6098, salary\_6098, contact\_6098;

Employee\_6098(**int** empID\_6098, String name\_6098)

{

**this**.empID\_6098 = empID\_6098;

**this**.name\_6098 = name\_6098;

}

**void** setEmpID\_6098(**int** empID\_6098)

{

**this**.empID\_6098 = empID\_6098;

}

**void** setName\_6098(String name\_6098)

{

**this**.name\_6098 = name\_6098;

}

**void** setDesignation\_6098(String designation\_6098)

{

**this**.designation\_6098 = designation\_6098;

}

**void** setSalary\_6098(String salary\_6098)

{

**this**.salary\_6098 = salary\_6098;

}

**void** setContact\_6098(String contact\_6098)

{

**this**.contact\_6098 = contact\_6098;

}

**int** getEmpID\_6098()

{

**return this**.empID\_6098;

}

String getName\_6098()

{

**return this**.name\_6098;

}

String getDesignation\_6098()

{

**return this**.designation\_6098;

}

String getSalary\_6098()

{

**return this**.salary\_6098;

}

String getContact\_6098()

{

**return this**.contact\_6098;

}

}

### EmpManager Class ::

**import** java.io.File;

**import** java.io.FileInputStream; **import** java.io.FileNotFoundException; **import** java.io.FileOutputStream; **import** java.io.IOException;

**import** java.io.ObjectInputStream; **import** java.io.ObjectOutputStream; **import** java.util.ArrayList; **import** java.util.List;

**import** java.util.Scanner;

**public class** EmpManager\_6098

{

List<Employee\_6098> l\_6098 = **new** ArrayList<>(); List<Employee\_6098> readList\_6098;

**void** addEmployee\_6098(Scanner s\_6098)

{



");

");

System.***out***.println(" -->> Enter the employee ID : "); **int** empid\_6098 = s\_6098.nextInt(); System.***out***.println(" -->> Enter the employee name :

String name\_6098 = s\_6098.next(); System.***out***.println(" -->> Enter the designation : "); String designation\_6098 = s\_6098.next(); System.***out***.println(" -->> Enter the salary : "); String sal\_6098 = s\_6098.next(); System.***out***.println(" -->> Enter the contact number :

String contact\_6098 = s\_6098.next();

Employee\_6098 e\_6098 = **new** Employee\_6098(empid\_6098, name\_6098);

e\_6098.setDesignation\_6098(designation\_6098); e\_6098.setSalary\_6098(sal\_6098); e\_6098.setContact\_6098(contact\_6098); l\_6098.add(e\_6098);

**for**(Employee\_6098 item\_6098 : l\_6098)

{

System.***out***.println(" >> Employee added successfully " + item\_6098.getEmpID\_6098() + " " + item\_6098.getName\_6098());

}

System.***out***.println(" >> Employee added successfully "

+ l\_6098);

}

**void** writeData\_6098(String name\_6098)

{

**if**(name\_6098 != **null** && !name\_6098.isEmpty())

{

**if**(l\_6098.size() > 0)

{

File f\_6098 = **new** File(name\_6098);

#### try

{

FileOutputStream(f\_6098);

FileOutputStream nFOS\_6098 = **new**

ObjectOutputStream nOOS\_6098 = **new**

ObjectOutputStream(nFOS\_6098);

nOOS\_6098.writeObject(l\_6098);

System.***out***.println(" >> Writing process completed successfully");

}

**catch**(FileNotFoundException e)

{

e.printStackTrace();

}

**catch**(IOException e)

{

e.printStackTrace();

}

}

#### else

{

System.***out***.println(" >> No employee exists.

Create some employee first");

}

}

#### else

{

");

}

}

System.***out***.println(" -->> Provide the file name :

**void** readEmpData\_6098(String name\_6098)

{

**if**(name\_6098 != **null** && !name\_6098.isEmpty())

{

File f\_6098 = **new** File(name\_6098);

**if**(f\_6098.exists())

{

#### try

{

FileInputStream(f\_6098);

FileInputStream fFIS\_6098 = **new**

ObjectInputStream fOIS\_6098 = **new**

ObjectInputStream(fFIS\_6098);

readList\_6098 = (List<Employee\_6098>)

fOIS\_6098.readObject();

successfully");

}

System.***out***.println(" >> Employee read

**catch**(FileNotFoundException e)

{

e.printStackTrace();

}

**catch**(IOException e)

{

e.printStackTrace();

}

**catch**(ClassNotFoundException e)

{

}

}

#### else

{

e.printStackTrace();

exists");

}

}

System.***out***.println(" >> File does not

}



**void** displayRecords\_6098(String name\_6098)

{

readEmpData\_6098(name\_6098);

**if**(readList\_6098.size() > 0)

{

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



System.***out***.println(" EmpID Name Designation Salary Contact");

**for**(Employee\_6098 item\_6098 : readList\_6098)

{

System.***out***.println(" "+item\_6098.getEmpID\_6098()+" "+item\_6098.getName\_6098()+" "+item\_6098.getDesignation\_6098()+" "+item\_6098.getSalary\_6098()+" "+item\_6098.getContact\_6098());

}

exists");

}

}

}

#### else

{

}

System.***out***.println(" >> Employee does not

### Launcher Class ::

**import** java.util.Scanner;

**public class** Launcher\_6098

{

Scanner s\_6098 = **new** Scanner(System.***in***);

**int** option\_6098;

**public static void** main(String args[])

{

Launcher\_6098 l\_6098 = **new** Launcher\_6098(); EmpManager\_6098 em\_6098 = **new** EmpManager\_6098(); l\_6098.mainMenu\_6098(l\_6098, em\_6098);

}

**void** mainMenu\_6098(Launcher\_6098 l\_6098, EmpManager\_6098 em\_6098)

{

System.***out***.println(""); System.***out***.println(" !! Main Menu !! "); System.***out***.println("1. Create Employee");

System.***out***.println("2. Write employee details"); System.***out***.println("3. Read employee details"); System.***out***.println("4. Display employee details"); System.***out***.println("5. Exit the menu");

option\_6098 = s\_6098.nextInt();

**switch**(option\_6098)

{

**case** 1:

System.***out***.println(" !! You are in create

employee area !! ");

em\_6098.addEmployee\_6098(s\_6098); mainMenu\_6098(l\_6098, em\_6098); **break**;

**case** 2:

System.***out***.println(" !! You are in write

employee area !! ");

em\_6098.writeData\_6098("test.txt"); mainMenu\_6098(l\_6098, em\_6098); **break**;

**case** 3:

System.***out***.println(" !! You are in read

employee area !! ");

em\_6098.readEmpData\_6098("test.txt"); mainMenu\_6098(l\_6098, em\_6098); **break**;

**case** 4:

System.***out***.println(" !! You are in display

employee area !! ");

em\_6098.displayRecords\_6098("test.txt"); mainMenu\_6098(l\_6098, em\_6098);

#### break;

**case** 5:

System.***out***.println(" !! Exited !! "); System.*exit*(0);

#### break;

}

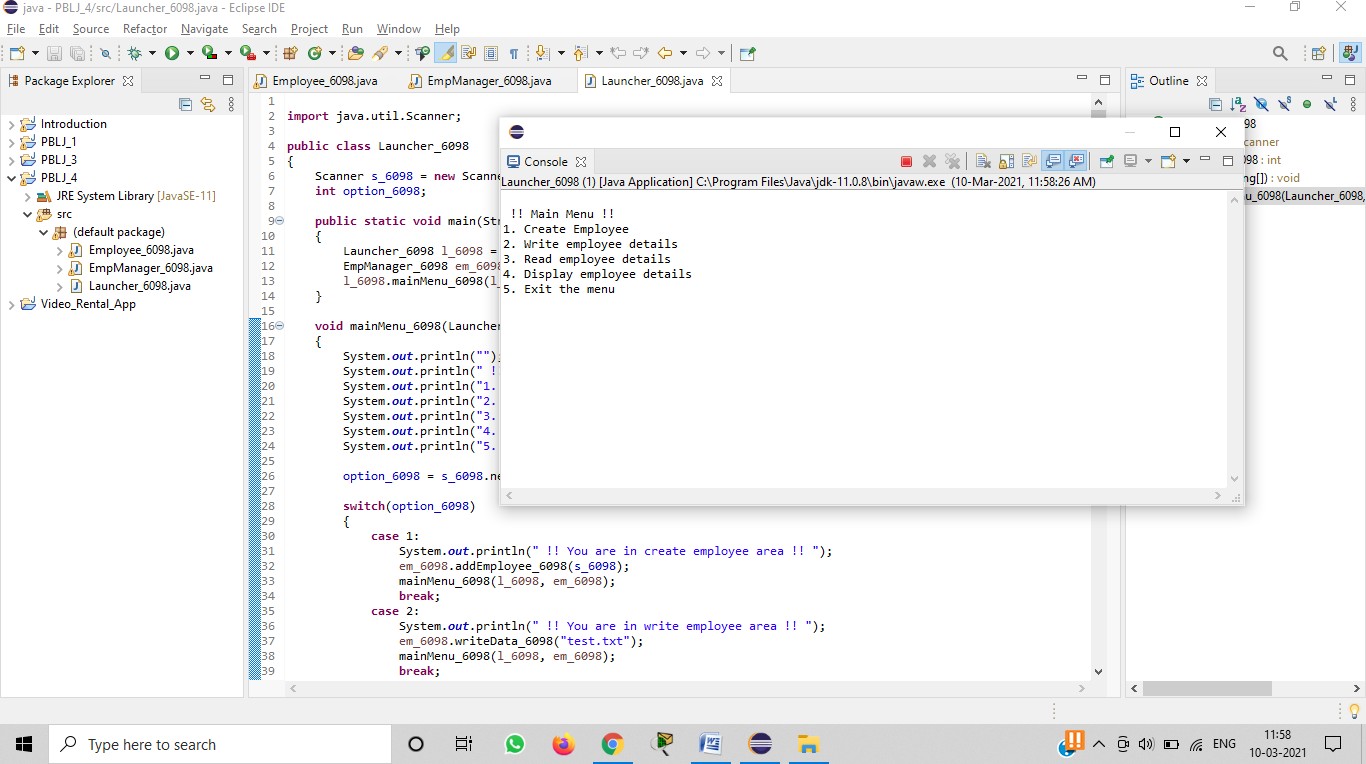
}

}

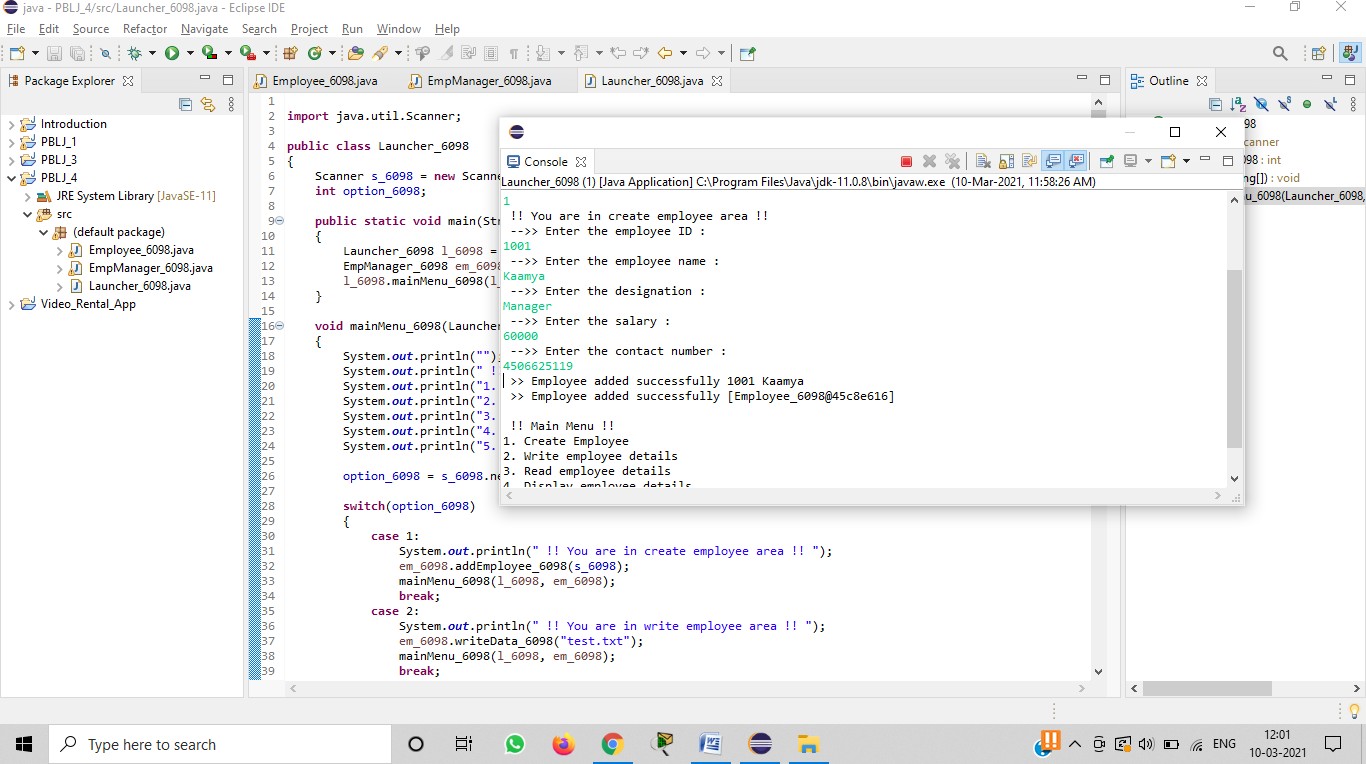
1. **Observations/Discussions(For applied/experimental sciences/materials based labs):** nil

## Percentage error (if any or applicable): nil

1. **Calculations/ Chemical Reactions / Theorems /Formulas used etc :** nil
2. **Result/Output/Writing Summary:** Output ::
   * Initial look –



* + Creating an employee –





Discover . Lea i 4. E s power.

# GRADE¥ D"

ACCREDITED UNIVERSITY

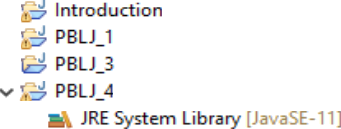
* + Writing data of employee –

@ K\_ s'+^

va Lau nc he r\_60g8.java [ ” 6 p”• Ou II i ne ” 6

a s .

" S c ann\* Launch er 6098 (1) [Java Applie ation] C:\ Pro9 ram F iies‘\Java\j dk-11.0.8\ bin‘\|avaw.ex e (10-Mar-2021, :'i8:26 AM)

» employee added su<<ess\*ully [zmployee\_sea s<seuiu]

,g))



cu )

„ „, ,;„pg....) » wiring process completed su<<essf«lly



; ,y(.. ?. Wnitv’vmployuv’dutails

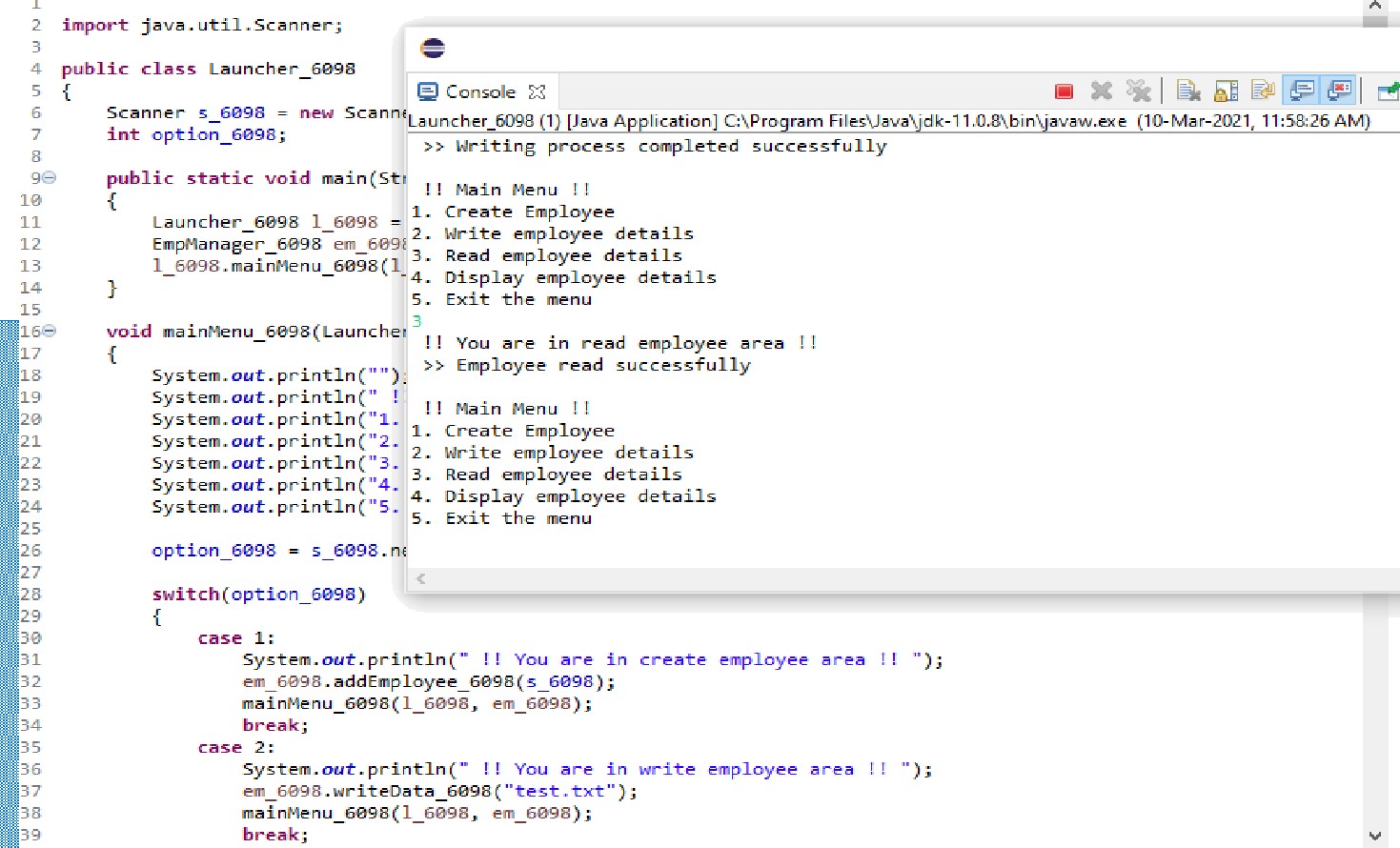




gfi Q Type here to sea rch

* + Reading data of employee –





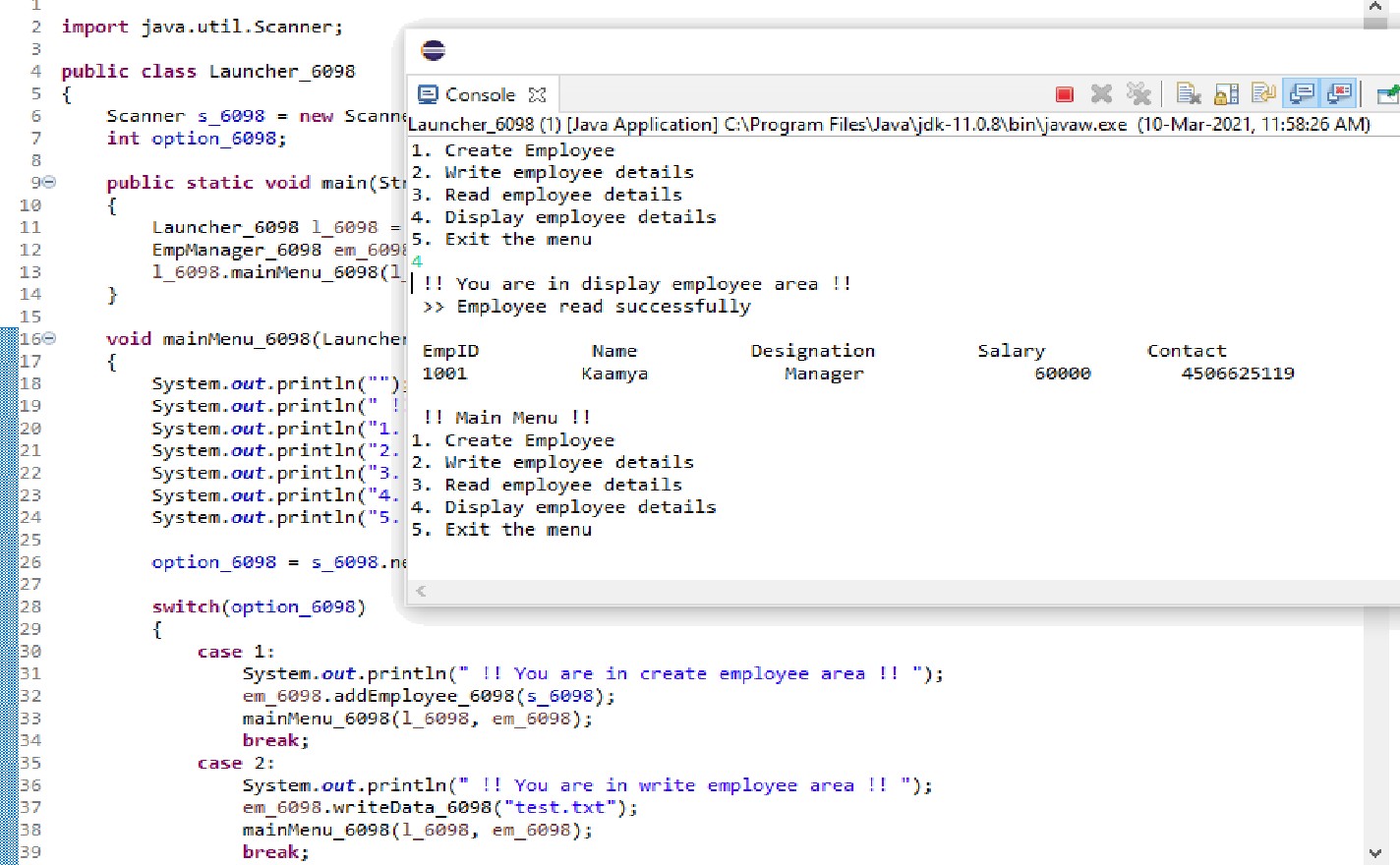
12:03



Discover . Lea i 4. E s power.

# GRADE¥ D"

ACCREDITED UNIVERSITY

* + Displaying employee details –
* cu c e F'efa ct c- NJ a--- ie at e e ai c h Pre e ct Pu a in cic <' Hei



Vid eo\_R ental\_Ap p

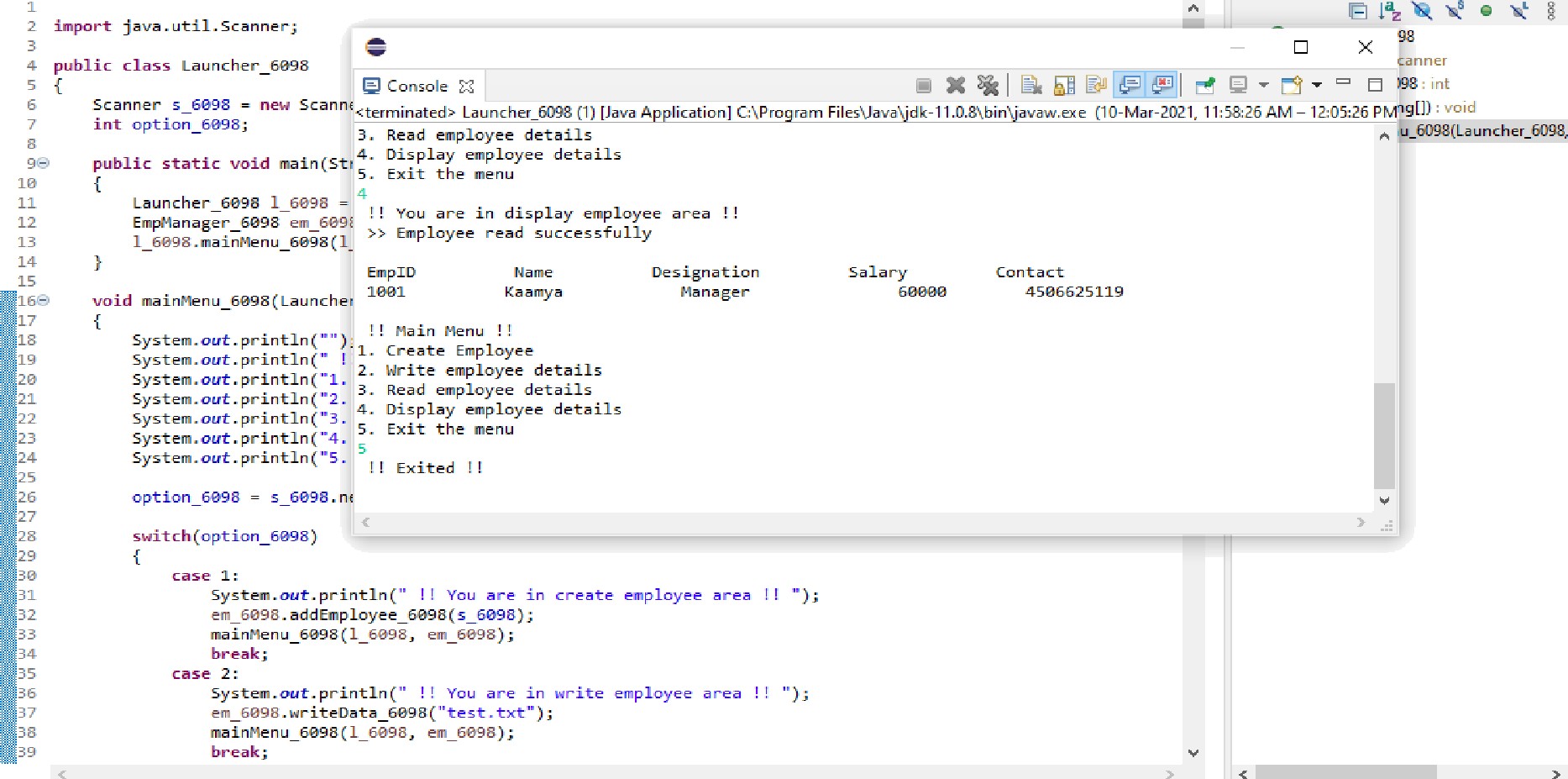
6 • & @ | 9’

ava D ” EN °—- Ou t line D ” 6



cu )

### Exiting –



fiB Q here to search



1. **Graphs (If Any): Image /Soft copy of graph paper to be attached here:** nil
2. **Learning outcomes (What I have learnt):**

### Learnt about the concept of files.

1. Learnt about the concept of serializable.
2. Learnt about file handling.
3. Learnt about implementation of file handling.
4. Learnt about the concept of array list and how to work with it.
5. **Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |