LAB INDEX

NAME: Vivek Kumar SUBJECTNAME: Project Based Learning in Java Lab

UID: 21BCS8129 SUBJECTCODE: 20CSP-314

SECTION: WM-20BCS-616/A

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Program** | **Date** | **Evaluation** | | | | **Sign** |
| **LW**  **(12)** | **VV**  **(10)** | **FW**  **(8)** | **Total**  **(30)** |
| 1 | Create an application to save the employee information using arrays. | 09-08-2022 |  |  |  |  |  |
| 2 | Design and implement a simple inventory control system for a small video rentalstore. | 23-08-2022 |  |  |  |  |  |
| 3 | Create a application to calculate interest for FDs, RDs based on certain conditions using inheritance. | 02-09-2022 |  |  |  |  |  |
| 4 | Create a program to show the usage of Sets of Collection interface. | 27-09-2022 |  |  |  |  |  |
| 5 | Create a program to set view of Keys from Java Hashtable. | 27-09-2022 |  |  |  |  |  |
| 6 | Write a Program to perform the basic operations like insert, delete, display and search in list. List contains String object items where these operations are to be performed. | 04-10-2022 |  |  |  |  |  |
| 7 | Create a menu based Java application with the following options.1.Add an Employee2.Display All3.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. | 14-10-2022 |  |  |  |  |  |
| 8 | Create a palindrome creator application for making a longest possible palindrome out of given input string. |  |  |  |  |  |  |
| 9 | Create a Servlet/ application with a facility to print any message on web browser. |  |  |  |  |  |  |
| 10 | Create JSP application for addition, multiplication and division. |  |  |  |  |  |  |

## CHANDIGARH UNIVERSITY

## UNIVERSITY INSTITUTE OF NGINEERING

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



|  |  |
| --- | --- |
| **Submitted By: Submitted To:**  Vivek Kumar(21BCS8129) Neeru Sharma(E12950) | |
| **Subject Name** | Project Based Learning in Java Lab |
| **Subject Code** | 20CSP-321 |
| **Branch** | Computer Science and Engineering |
| **Semester** | 5th |

**Experiment - 7**

**Student Name: Vivek Kumar UID: 21BCS8129**

**Branch: BE-CSE(LEET) Section/Group:20BCS-WM-616/A**

**Semester: 5th Date of Performance: 14/10/2022**

**Subject Name:** **Project Based Learning in Java Lab Subject Code: 20CSP-321**

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

Create a menu-based Java application with the following options.1. Add an Employee2.Display All3.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.

**2. Task to be done/ Which logistics used:**

Write the program to create an application to perform a File manipulation.

**3. Software Requirements (For programming-based labs):**

* JDK-8 or any
* Eclipse-IDE for Java

**4. Steps for experiment/practical/Code:**

package unit2;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.Serializable;

import java.util.ArrayList;

import java.util.Scanner;

@SuppressWarnings("serial")

class Employee implements Serializable {

int id;

String name;

float salary;

long contact\_no;

String email\_id;

public Employee(

int id,

String name,

float salary,

long contact\_no,

String email\_id

) {

this.id = id;

this.name = name;

this.salary = salary;

this.contact\_no = contact\_no;

this.email\_id = email\_id;

}

public String toString() {

return (

"Employee Details :\n" +

" ID: " +

this.id +

" Name: " +

this.name +

" Salary: " +

this.salary +

" Contact No: " +

this.contact\_no +

" Email-id: " +

this.email\_id

);

}

}

public class WorkSheet7 {

static void display(ArrayList<Employee> al) {

System.out.println("\n--------------Employee List---------------\n");

System.out.println(

String.format(

"%-10s%-15s%-10s%-20s%-10s",

"ID",

"Name",

"salary",

"contact-no",

"Email-Id"

)

);

for (Employee e : al) {

System.out.println(

String.format(

"%-10s%-15s%-10s%-15s%-10s",

e.id,

e.name,

e.salary,

e.contact\_no,

e.email\_id

)

);

}

}

@SuppressWarnings("unchecked")

public static void main(String[] args) {

int id;

String name;

float salary;

long contact\_no;

String email\_id;

Scanner sc = new Scanner(System.in);

ArrayList<Employee> al = new ArrayList<Employee>();

File f = null;

FileInputStream fis = null;

ObjectInputStream ois = null;

FileOutputStream fos = null;

ObjectOutputStream oos = null;

try {

f =

new File(

"R:\\VnjVibhash\\Assignments\\CU-Assignments\\5th Sem\\Java\\JavaLab\\src\\EmployeeDataList.txt"

);

if (f.exists()) {

fis = new FileInputStream(f);

ois = new ObjectInputStream(fis);

al = (ArrayList<Employee>) ois.readObject();

}

} catch (Exception exp) {

System.out.println(exp);

}

do {

System.out.println(

"\n\*\*\*\*\*\*\*\*\*Welcome to the Employee Management System\*\*\*\*\*\*\*\*\*\*\n"

);

System.out.println(

"1). Add Employee to the DataBase\n" +

"2). Search for Employee\n" +

"3). Edit Employee details\n" +

"4). Delete Employee Details\n" +

"5). Display all Employees working in this company\n" +

"6). EXIT\n"

);

System.out.println("Enter your choice : ");

int ch = sc.nextInt();

switch (ch) {

case 1:

System.out.println("\nEnter the following details to ADD list:\n");

System.out.println("Enter ID :");

id = sc.nextInt();

System.out.println("Enter Name :");

name = sc.next();

System.out.println("Enter Salary :");

salary = sc.nextFloat();

System.out.println("Enter Contact No :");

contact\_no = sc.nextLong();

System.out.println("Enter Email-ID :");

email\_id = sc.next();

al.add(new Employee(id, name, salary, contact\_no, email\_id));

display(al);

break;

case 2:

System.out.println("Enter the Employee ID to search :");

id = sc.nextInt();

int i = 0;

for (Employee e : al) {

if (id == e.id) {

System.out.println(e + "\n");

i++;

}

}

if (i == 0) {

System.out.println(

"\nEmployee Details are not available, Please enter a valid ID!!"

);

}

break;

case 3:

System.out.println("\nEnter the Employee ID to EDIT the details");

id = sc.nextInt();

int j = 0;

for (Employee e : al) {

if (id == e.id) {

j++;

do {

int ch1 = 0;

System.out.println(

"\nEDIT Employee Details :\n" +

"1). Employee ID\n" +

"2). Name\n" +

"3). Salary\n" +

"4). Contact No.\n" +

"5). Email-ID\n" +

"6). GO BACK\n"

);

System.out.println("Enter your choice : ");

ch1 = sc.nextInt();

switch (ch1) {

case 1:

System.out.println("\nEnter new Employee ID:");

e.id = sc.nextInt();

System.out.println(e + "\n");

break;

case 2:

System.out.println("Enter new Employee Name:");

e.name = sc.nextLine();

System.out.println(e + "\n");

break;

case 3:

System.out.println("Enter new Employee Salary:");

e.salary = sc.nextFloat();

System.out.println(e + "\n");

break;

case 4:

System.out.println("Enter new Employee Contact No. :");

e.contact\_no = sc.nextLong();

System.out.println(e + "\n");

break;

case 5:

System.out.println("Enter new Employee Email-ID :");

e.email\_id = sc.next();

System.out.println(e + "\n");

break;

case 6:

j++;

break;

default:

System.out.println(

"\nEnter a correct choice from the List :"

);

break;

}

} while (j == 1);

}

}

if (j == 0) {

System.out.println(

"\nEmployee Details are not available, Please enter a valid ID!!"

);

}

break;

case 4:

System.out.println(

"\nEnter Employee ID to DELETE from the Databse :"

);

id = sc.nextInt();

int k = 0;

try {

for (Employee e : al) {

if (id == e.id) {

al.remove(e);

display(al);

k++;

}

}

if (k == 0) {

System.out.println(

"\nEmployee Details are not available, Please enter a valid ID!!"

);

}

} catch (Exception ex) {

System.out.println(ex);

}

break;

case 5:

try {

al = (ArrayList<Employee>) ois.readObject();

} catch (ClassNotFoundException e2) {

System.out.println(e2);

} catch (Exception e2) {

System.out.println(e2);

}

display(al);

break;

case 6:

try {

fos = new FileOutputStream(f);

oos = new ObjectOutputStream(fos);

oos.writeObject(al);

} catch (IOException e1) {

e1.printStackTrace();

} catch (Exception e2) {

e2.printStackTrace();

} finally {

try {

fis.close();

ois.close();

fos.close();

oos.close();

} catch (Exception e1) {

e1.printStackTrace();

}

}

System.out.println(

"\nYou have chosen EXIT !! Saving Files and closing the tool."

);

sc.close();

System.exit(0);

break;

default:

System.out.println("\nEnter a correct choice from the List :");

break;

}

} while (true);

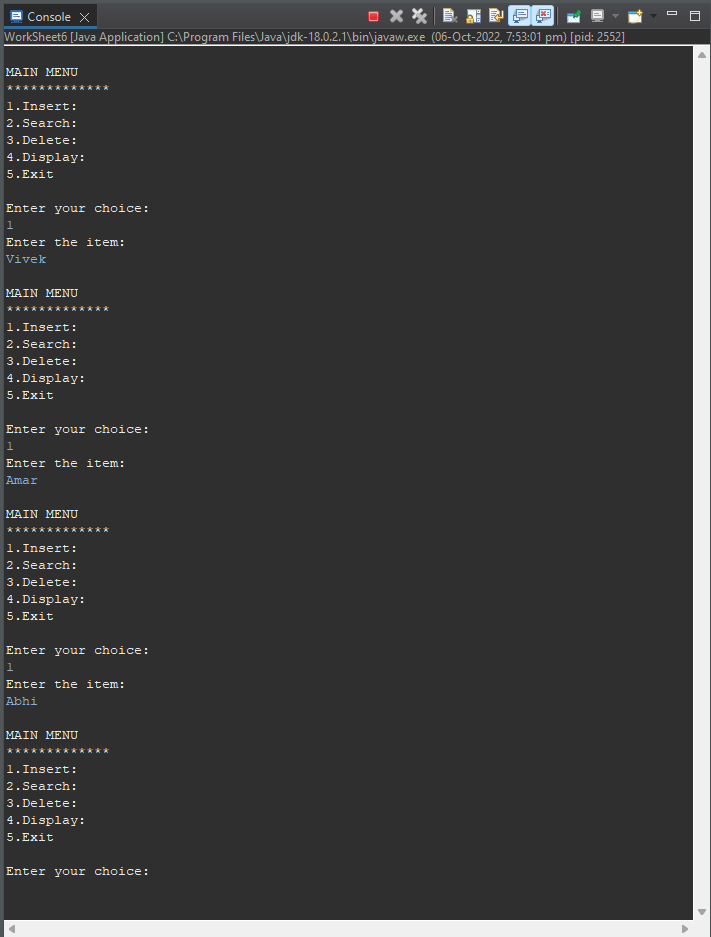
}

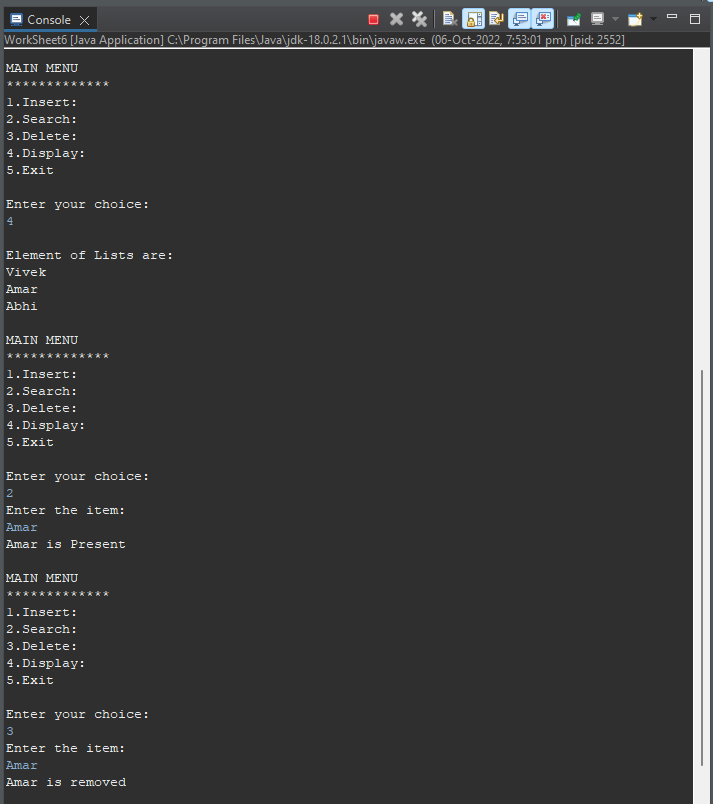
}

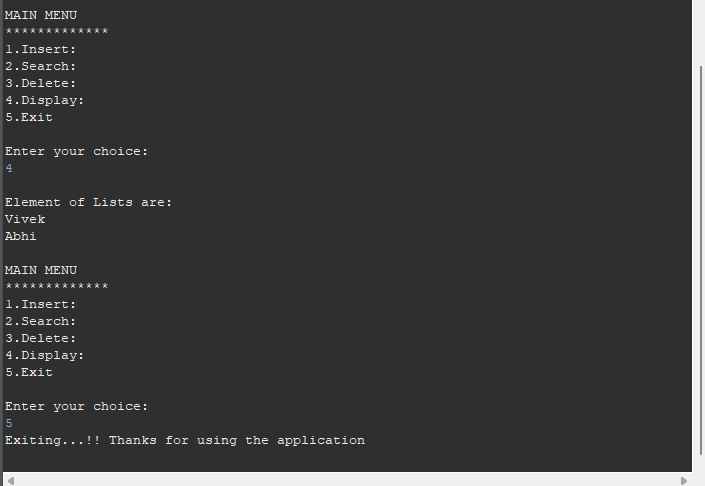
**5. Observations/Discussions/ Complexity Analysis:**

Here we have created the List, and performed all the operation of list such as insertion, searching, deletion, and traversal.

**6. Result/Output/Writing Summary:**







**Learning outcomes (What I have learnt):**

**1.** Learnt while loop.

**2.** List manipulation concept understood.

**3.** Created list and performed all operation of list.

**4.** Learnt the concept of switch concept.

**5.** Learnt concept of inbuilt function in list.

**Evaluation Grid (To be created per the faculty's SOP and Assessment guidelines):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. | Worksheet completion including writing learning objectives/Outcomes.  (To be submitted at the end of the day). |  |  |
| 2. | Post-Lab Quiz Result. |  |  |
| 3. | Student Engagement in  Simulation/Demonstration/Performance and Controls/Pre-Lab Questions. |  |  |
|  | Signature of Faculty (with Date): | Total Marks Obtained: |  |