**Experiment Title 2.4**

**Student Name: Sahul Kumar Parida UID: 20BCS4919**

**Branch: CSE Section/Group: WM-904/B**

**Semester: 5th**

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

**Subject Code: 20CSP-321**

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

Employee Management System

Create a menu based Java application with the following options.

1. Add an Employee

2. 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.

**2. Software/Hardware Requirements:**

# Windows

# BlueJ (Java IDE)

# JRE (Java Runtime Environment)

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

Employee.java

import java.io.Serializable;

public class Employee implements Serializable

{

private int id;

private String name;

private double salary;

private int age;

public int getId()

{

return id;

}

public void setId(int id)

{

this.id = id;

}

public String getName()

{

return name;

}

@Override

public String toString()

{

return "Employee [id=" + id + ", name=" + name + ", salary=" + salary

+ ", age=" + age + "]";

}

public Employee(int id, String name, double salary, int age)

{

super();

//this.id = id;

setId(id);

//this.name = name;

setName(name);

//this.salary = salary;

setSalary(salary);

//this.age = age;

setAge(age);

}

public void setName(String name)

{

this.name = name;

}

public double getSalary()

{

return salary;

}

public void setSalary(double salary)

{

this.salary = salary;

}

public int getAge()

{

return age;

}

public void setAge(int age)

{

this.age = age;

}

}

EmployeeManager.java

import java.util.\*;

public class EmployeeManager

{

int menuChoice;

Scanner objectScanner= new Scanner(System.in);

List<Employee> objectEmployeeList= new LinkedList<Employee>();

public static void main()

{

EmployeeManager objectManager= new EmployeeManager();

objectManager.displayMenu();

while(objectManager.menuChoice!=3)

{

objectManager.displayMenu();

}

}

void displayMenu()

{

System.out.println("Main Menu");

System.out.println("1.Add an Employee");

System.out.println("2.Display All");

System.out.println("3.Exit");

menuChoice=objectScanner.nextInt();

try

{

switch(menuChoice)

{

case 1:

getEmployeeInformationFromConsole();

break;

case 2:

displayFromList(objectEmployeeList);

break;

case 3:

System.out.println("Exiting the program");

System.exit(0);

}

}

catch(InputMismatchException e)

{

System.out.println("Exception is----->"+e);

}

}

void getEmployeeInformationFromConsole()

{

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

try

{

int id=objectScanner.nextInt();

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

String name=objectScanner.next();

System.out.println ("Enter Employee Age:");

int age=objectScanner.nextInt();

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

double salary= objectScanner.nextDouble();

objectEmployeeList.add(new Employee(id, name, salary, age));

}

catch(InputMismatchException e)

{

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

}

}

String displayFromList(List<Employee> objectList)

{

String status="";

try{

Iterator objectIterator= objectList.iterator();

while(objectIterator.hasNext())

{

System.out.println (""+objectIterator.next());

}

status="Displayed Successfully";

return status;

}

catch(Exception e)

{

status="Failure";

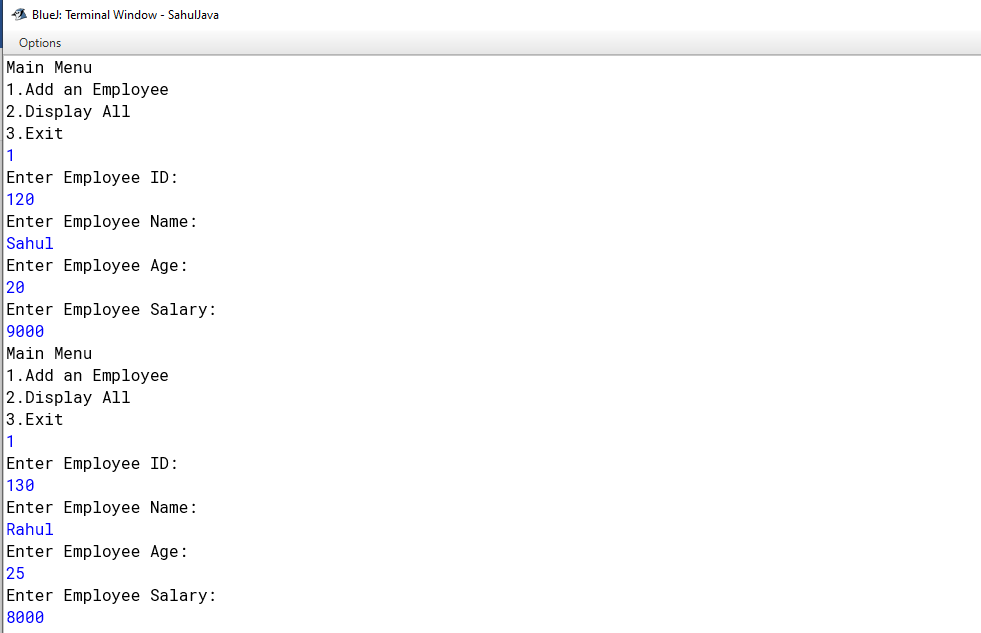
return status;

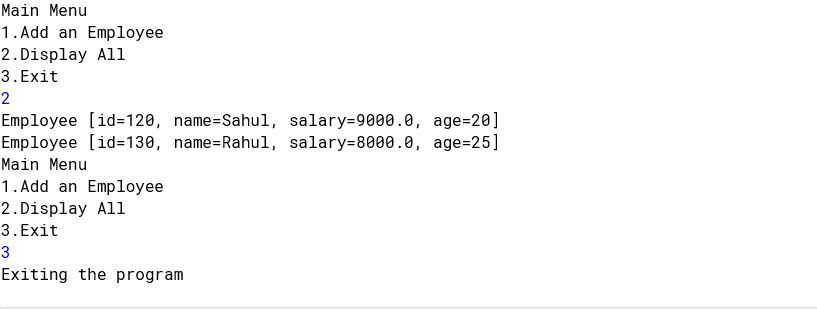
}

}

}

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





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

1. **Switch-Case Statements**
2. **Java Serialization**
3. **ArrayList**