

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Experiment No: 2.4

**Student Name:** Sachin Kumar Singh

**UID:** 21BCS9217

**Branch:** BE-CSE

**Section/Group:** 21BCS-CC-646 (B)

**Semester:** 6<sup>th</sup>

**Date of Performance:** 04/02/2024

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

**Subject Code:** 21CSH-319

**Aim:** Create a program to perform writing and reading operations in a text file.

### Objective:

1. To learn about IO.
2. To learn about reading and writing file in java.

**Input/Apparatus Used:** IntelliJ / VS Code.

### Procedure/Algorithm/Pseudocode

1. Define a record named Employee with fields for id, name, age, and salary.
2. Override the toString() method in the Employee record to return a formatted string representation.
3. Define a class named EmployeeManagement:
  - a. Define a method named addAnEmployee to add a new employee to the file.
  - b. Open a BufferedWriter to append data to the "res/employee.txt" file.
  - c. Write the string representation of the employee followed by a newline character.
  - d. Close the BufferedWriter.
4. Define a method named displayAll to display all employees from the file.
  - a. Create an ArrayList<Employee> to store employees.
  - b. Open a BufferedReader to read data from the "res/employee.txt" file.
  - c. Read each line from the file.
  - d. Split the line into parts using space as the delimiter.
  - e. Parse each part to construct an Employee object and add it to the



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

- ArrayList.
  - f. Close the BufferedReader.
  - g. Catch IOException and NumberFormatException, and print an error message if reading or parsing fails.
  - h. Iterate over the ArrayList<Employee> and display each employee's details.
5. Define a method named run to execute the main menu loop:
- a. Loop indefinitely.
  - b. Display the main menu options.
  - c. Prompt the user to enter their choice.
  - d. Use a switch statement to handle the user's choice:
  - e. Case 1: Prompt for and add a new employee.
  - f. Case 2: Display all employees.
  - g. Case 3: Exit the program.
  - h. Default: Display a message for a wrong choice.
6. Define the Main class:
- a. Define the main method:
  - b. Create a Scanner to read user input.
  - c. Create an instance of EmployeeManagement.
  - d. Call the run method of EmployeeManagement with the Scanner.
  - e. Close the Scanner.

## Code:

```
package University.Java_Using_Project.Experiment7;

import java.io.*;

import java.util.ArrayList;

import java.util.Scanner;

record Employee(int id, String name, int age, double salary) {

    @Override
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
public int id() {  
    return id;  
}  
  
@Override  
public String name() {  
    return name;  
}  
  
@Override  
public int age() {  
    return age;  
}  
  
@Override  
public double salary() {  
    return salary;  
}  
  
@Override  
public String toString() {  
    return id + " " + name + " " + age + " " + salary;  
}  
}  
  
class EmployeeManagement {  
  
    public void addAnEmployee(Employee employee) {
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
try (BufferedWriter writer = new BufferedWriter(new
FileWriter("res/employee.txt", true))) {

    writer.write(employee.toString());

    writer.newLine();

} catch (IOException e) {

    System.out.println("Error Occurred!!!");

}

}

public void displayAll() {

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

    try (BufferedReader reader = new BufferedReader(new
FileReader("res/employee.txt"))) {

        String data;

        while ((data = reader.readLine()) != null) {

            String[] str = data.split(" ");

            int id = Integer.parseInt(str[0]);

            String name = str[1];

            int age = Integer.parseInt(str[2]);

            double salary = Double.parseDouble(str[3]);

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

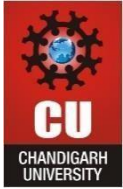
        }

    } catch (IOException | NumberFormatException e) {

        System.out.println("Error Occurred: " + e.getMessage());

    }

}
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
    }

    for (Employee employee : employees) {

        System.out.println("Name: " + employee.name() +
            " UID: " + employee.id() +
            " Age: " + employee.age() +
            " Salary: " + employee.salary());

    }

}

void run(Scanner in) {

    while (true) {

        System.out.println("\nEnter 1 for Add an Employee");
        System.out.println("Enter 2 for Display All");
        System.out.println("Enter 3 for Exit\n");
        System.out.print("Enter your choice:");

        int choice = in.nextInt();

        switch (choice) {

            case 1: {

                System.out.print("Enter the Employee Id: ");

                int id = in.nextInt();

                in.nextLine();

                System.out.print("Enter the Employee Name: ");

                String name = in.nextLine();

                System.out.print("Enter the Employee Age: ");
```



```
        int age = in.nextInt();

        System.out.print("Enter the Employee Salary: ");

        double salary = in.nextDouble();

        addAnEmployee(new Employee(id, name, age, salary));
    }

    break;

case 2: {

    displayAll();

}

break;

case 3: {

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

    System.exit(0);

}

break;

default: {

    System.out.println("Wrong Choice!!!");

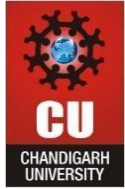
}break;

}

}

}
```

```
public class Main {
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
public static void main(String[] args) {  
  
    Scanner in = new Scanner(System.in);  
  
    EmployeeManagement system = new EmployeeManagement();  
  
    system.run(in);  
  
    in.close();  
  
}  
  
}
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Result/Output:

```
Enter 1 for Add an Employee
Enter 2 for Display All
Enter 3 for Exit

Enter your choice:1
Enter the Employee Id: 9217
Enter the Employee Name: Sachin
Enter the Employee Age: 21
Enter the Employee Salary: 200000
```

```
Enter 1 for Add an Employee
Enter 2 for Display All
Enter 3 for Exit

Enter your choice:2
Name: Sachin UID: 9217 Age: 20 Salary: 200000.0
Name: Sachin UID: 9217 Age: 21 Salary: 200000.0
Name: Ayush UID: 9202 Age: 21 Salary: 200000.0
Name: Kalpana UID: 9195 Age: 20 Salary: 50000.0
Name: Sarbjeet UID: 9194 Age: 24 Salary: 150000.0
Name: Nisha UID: 9166 Age: 20 Salary: 50000.0
```

```
Main.java  employee.txt x
1  9217 Sachin 20 200000.0
2  9217 Sachin 21 200000.0
3  9202 Ayush 21 200000.0
4  9195 Kalpana 20 50000.0
5  9194 Sarbjeet 24 150000.0
6  9166 Nisha 20 50000.0
```