

ABHISHEK DHAR
23201
E-10

01

classmate

Date

Page

ASSIGNMENT - 08

Title → File Handling

Aim → Implement a program for maintaining a database of student records using files. Student has Student-id, Roll-no, Class, marks and address. Display data for few students (5)

- 1) Create database
- 2) Display database
- 3) Delete records
- 4) Update records
- 5) Search records.

Theory → File class from java.io package allows us to create an object, specify filename and directory.

Eg → `File my-file = new File("my-file.txt");`

`Obj.delete();` lets us delete a file object.

We can use the `FileWriter` class together with its `write()` method to write some text in the above file.

23201

```
FileWriter fw1 = new FileWriter("my-file");
```

After, we are done writing, we must close the file writer with the obj.close() method.

```
fw1.write("hello World");
```

// writes the text "hello World" in newly created file.

```
fw1.close();
```

// closes filewriter

We can use the scanner class to read the contents of the textfile.

```
Scanner  
Scanner Reader = new Scanner("my-file");  
while (Reader.hasNextLine())  
{  
    System.out.println(Reader.nextLine());  
}
```

Output →

Hello World

23/201

Reads previous data written by FileWriter.
We can close the Scanner object using
obj.close() method.

Working →

A class called student has the variables
ID, Roll-no, class, address and marks
also has a default constructor which sets
these default values.

Inside the main method, an loop asks
for input of 5 ~~inputs~~ students (checking
takes place to prevent duplicate id)

A new file f1 is created called "Assignments
outputs.txt" and this data of all 5
students is written in that file.

A menu driven program asks the user
for operation to be performed.

1) Clear all data →

A new file is created of the exact
same name after deleting f1.

2) Delete one record →

23201

Scanner scans the file f_1 till it encounters the entered ID. It then skips this data and continues ahead, and copies into f_2 .
Delete R_1

3) Modifies records →

Scanner scans file f_1 till it encountered the entered ID, it skips this data and writes new data, then continues ahead and write into f_2 , Delete f_1 , Rename f_2 .

4) Search Records →

Scanner scans file f_1 till it encounters the entered ID, it prints this data in the terminal & continues ahead.

* Class Diagram →

Student	Assignment 8
String Student_ID; String Roll-no; String class; String address; double marks; Student()	Psum(String[] args)

Test case →

Data:

ID : 1, 2, 3, 4, 5

Roll-no : 1, 2, 3, 4, 5

Class : 1, 2, 3, 4, 5

Address : 1, 2, 3, 4, 5

Marks : 1, 2, 3, 4, 5

In file : All data of 5 students
2 → Enter ID : 2

Infile : Data of student 2 is deleted

3 → Enter ID : 3

New data : 100, 100, 100, 100

Infile : All data (3s are replaced by 100)

4 → Enter ID : 3

100, 100, 100, 100 } data displayed
in terminal

1 → All data in file deleted.