Assignment No:8

```
import java.io.*;
import java.util.*;

 class StudentRecords
{
 static BufferedReader br = new BufferedReader(new  InputStreamReader(System.in)) ;
    // Method to add records
 public void addRecords() throws IOException {
 PrintWriter pw = new PrintWriter(new BufferedWriter(new FileWriter("st.txt",    true)));
  String studentname, address;
  int studentid, rollno, studentClass;
  float marks;
  boolean addMore = false;
        String s;
      do {
          try {
              System.out.print("\nEnter Student Name: ");
              studentname = br.readLine();
              System.out.print("Student Id: ");
              studentid = Integer.parseInt(br.readLine());
              System.out.print("Roll no: ");
              rollno = Integer.parseInt(br.readLine());
              System.out.print("Address: ");
              address = br.readLine();
              System.out.print("Class: ");
              studentClass = Integer.parseInt(br.readLine());
```

```
            System.out.print("Marks: ");
            marks = Float.parseFloat(br.readLine());
            pw.println(studentname + " " + studentid + " " + rollno + " " + address + " " +
studentClass + " " + marks);
           System.out.println("\nRecord added successfully!");
         } catch (NumberFormatException e) {
            System.out.println("Invalid input. Please enter the correct data type.");
            continue;
         }
       System.out.print("\nDo you want to add more records? (y/n): ");
        s = br.readLine();
        addMore = s.equalsIgnoreCase("y");
       } while (addMore);
    pw.close();
    showMenu();
  }

  // Method to read records
  public void readRecords() throws IOException {
  try (BufferedReader file = new BufferedReader(new FileReader("st.txt")))
  {
       String name;
        while ((name = file.readLine()) != null) {
            System.out.println(name);
         }
    } catch (FileNotFoundException e) {
        System.out.println("ERROR: File not Found!");
    }
    showMenu();
}

// Search records by student id
public void searchRecords() throws IOException {
    try (BufferedReader file = new BufferedReader(new FileReader("st.txt"))) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the ID of the student you want to search: ");
        String searchId = sc.next();

        String record;
        boolean found = false;
        while ((record = file.readLine()) != null) {
            String[] line = record.split(" ");
            if (searchId.equalsIgnoreCase(line[1])) {
                System.out.println("Record found: " + record);
                found = true;
                break;
            }
        }
        if (!found) {
            System.out.println("Record not found.");
        }
    } catch (FileNotFoundException e) {
        System.out.println("ERROR: File not Found!");
    }
    showMenu();
    }

    // Delete records by student name
    public void deleteRecords() throws IOException {
        try (BufferedReader file1 = new BufferedReader(new FileReader("st.txt"));
             PrintWriter pw = new PrintWriter(new BufferedWriter(new FileWriter("st1.txt",
true)))) {

            Scanner sc = new Scanner(System.in);
            System.out.print("Enter the name of the student you want to delete: ");
            String searchname = sc.next();

            String record;
            boolean found = false;
            while ((record = file1.readLine()) != null) {
                String[] line = record.split(" ");
                if (!searchname.equalsIgnoreCase(line[0])) {
                    pw.println(record);
                } else {
                    System.out.println("Record found and deleted.");
                    found = true;
                }
            }
          if (!found)
           {
                System.out.println("Record not found.");
            }
        } catch (FileNotFoundException e) {
            System.out.println("ERROR: File not Found!");
        }
        replaceOldFile();
        showMenu();
    }

    // Update student records
    public void updateRecords() throws IOException {
        try (BufferedReader file1 = new BufferedReader(new FileReader("st.txt"));
        PrintWriter pw = new PrintWriter(new BufferedWriter(new
FileWriter("st1.txt",true))))
    {
           Scanner sc = new Scanner(System.in);
            System.out.print("Enter the name of the student you want to update: ");
            String searchname = sc.next();
            String record;
            boolean found = false;
            while ((record = file1.readLine()) != null) {
                String[] line = record.split(" ");
                if (!searchname.equalsIgnoreCase(line[0])) {
                    pw.println(record);
                } else {
                    System.out.println("Record found.");
                    System.out.print("Enter updated marks: ");
                    float updatedMarks = Float.parseFloat(sc.next());
                    pw.println(line[0] + " " + line[1] + " " + line[2] + " " + line[3] + " " + line[4]
+ " " + updatedMarks);
```

```java
                found = true;
            }
        }
        if (!found) {
            System.out.println("Record not found.");
        }
    } catch (FileNotFoundException e) {
        System.out.println("ERROR: File not Found!");
    }
    replaceOldFile();
    showMenu();
}

// Replace old file with updated file
private void replaceOldFile() {
    File delName = new File("st.txt");
    File oldName = new File("st1.txt");
    if (delName.delete()) {
        oldName.renameTo(delName);
    }
}

// Clear all records
public void clear(String filename) throws IOException {
    PrintWriter pw = new PrintWriter(new BufferedWriter(new FileWriter(filename)));
    pw.close();
    System.out.println("All Records cleared successfully!");
```

```
        showMenu();
    }

    // Menu to interact with the user
    public void showMenu() throws IOException
{
        System.out.println("****You have following operations to perform on
        File****");
        System.out.println("1. Add Records\n2. Display Records\n3. Clear All Records\n4.
        Search Records\n5. Delete Records\n6. Update Records\n7. Exit");
        System.out.print("Enter your choice: ");
        int choice = Integer.parseInt(br.readLine());

        switch (choice)
         {
         case 1 : addRecords();
                  break;
         case 2 : readRecords();
                 break;
          case 3 :clear("st.txt");
                 break;
          case 4 : searchRecords();
                 break;
          case 5 :deleteRecords();
                 break;
          case 6 : updateRecords();
                 break;
          case 7 : System.exit(0);
```

```
               default : System.out.println("Invalid choice!");
                           showMenu();
            }
    }

    public static void main(String[] args) throws IOException {
        StudentRecords call = new StudentRecords();
        call.showMenu();
    }
}
```

Output

```
****You have following operations to perform on File****
1. Add Records
2. Display Records
3. Clear All Records
4. Search Records
5. Delete Records
6.Update Records
7. Exit
Enter your choice: 1
Enter Student Name: poonam
Student Id: 1
Roll no: 1
Address: nashik
Class: 7
Marks: 99
Record added successfully!
```

```
Do you want to add more records? (y/n): y
Enter Student Name: komal
Student Id: 2
Roll no: 2
Address: pune
Class: 8
Marks: 90
Record added successfully!
Do you want to add more records? (y/n): n

 ****You have following operations to perform on File****
1. Add Records
2. Display Records
3. Clear All Records
4. Search Records
5. Delete Records
6.Update Records
7. Exit
Enter your choice: 2
poonam 1 1 nashik 7 99.0
komal 2 2 pune 8 90.0

****You have following operations to perform on File****
1. Add Records
2. Display Records
3. Clear All Records
4. Search Records
```

```
5. Delete Records
6.Update Records
7. Exit
Enter your choice: 4
Enter the ID of the student you want to search: 1
Record found: poonam 1 1 nashik 7 99.0
****You have following operations to perform on File****
1. Add Records
2. Display Records
3. Clear All Records
4. Search Records
5. Delete Records
6.Update Records
7. Exit
Enter your choice: 5
Enter the name of the student you want to delete: komal
Record found and deleted.


****You have following operations to perform on File****
1. Add Records
2. Display Records
3. Clear All Records
4. Search Records
5. Delete Records
6.Update Records
7. Exit
```

```
Enter your choice: 2
poonam 1 1 nashik 7 99.0

****You have following operations to perform on File****
1. Add Records
2. Display Records
3. Clear All Records
4. Search Records
5. Delete Records
6.Update Records
7. Exit
Enter your choice: 7
```