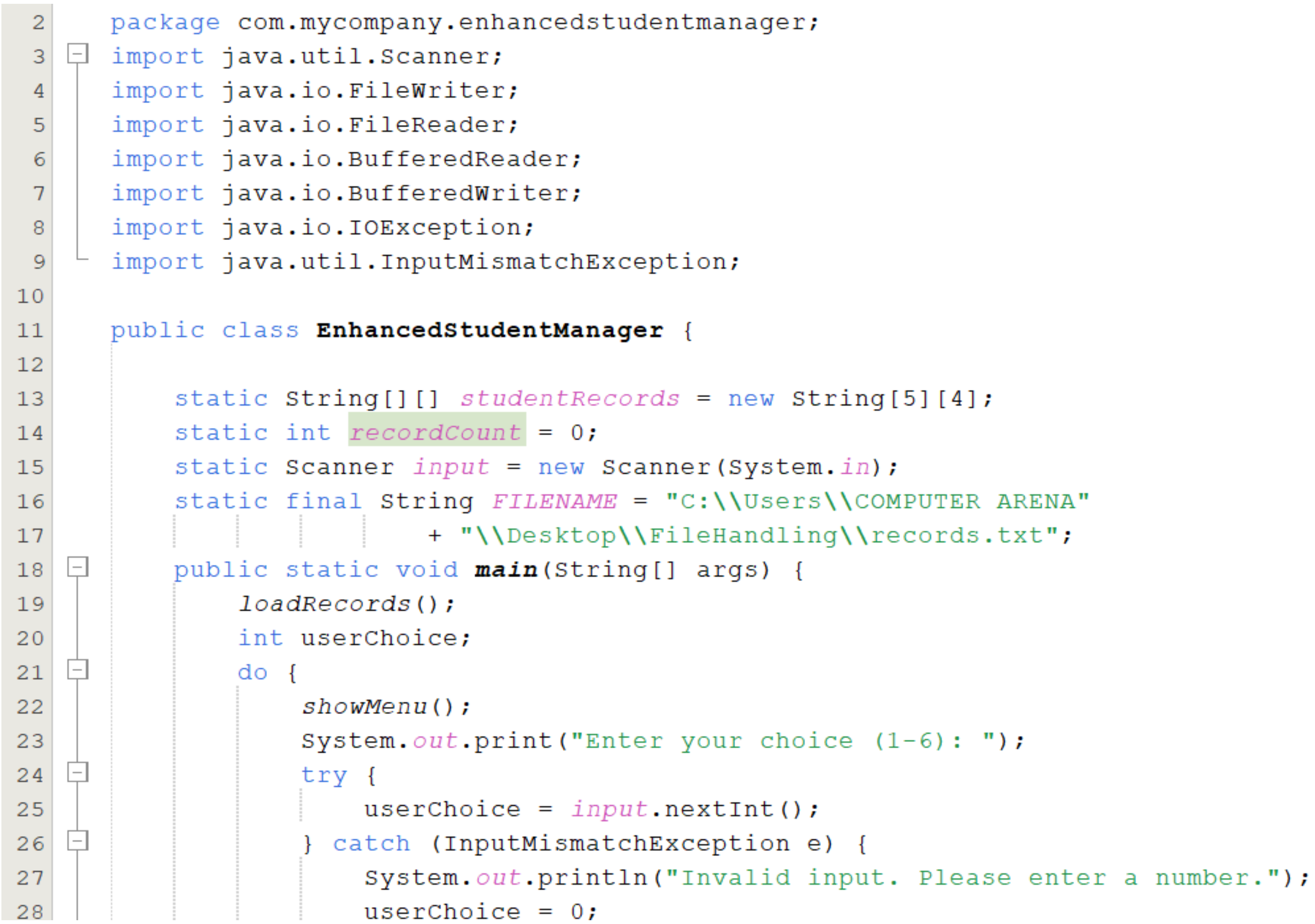
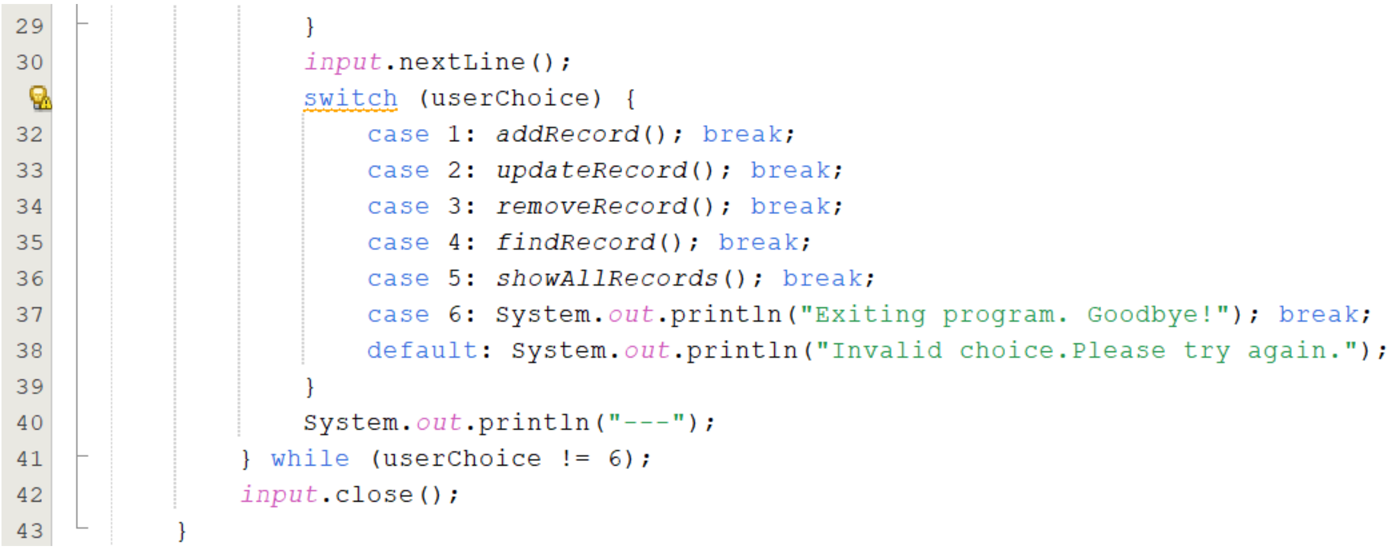
ENHANCED STUDENT MANAGER

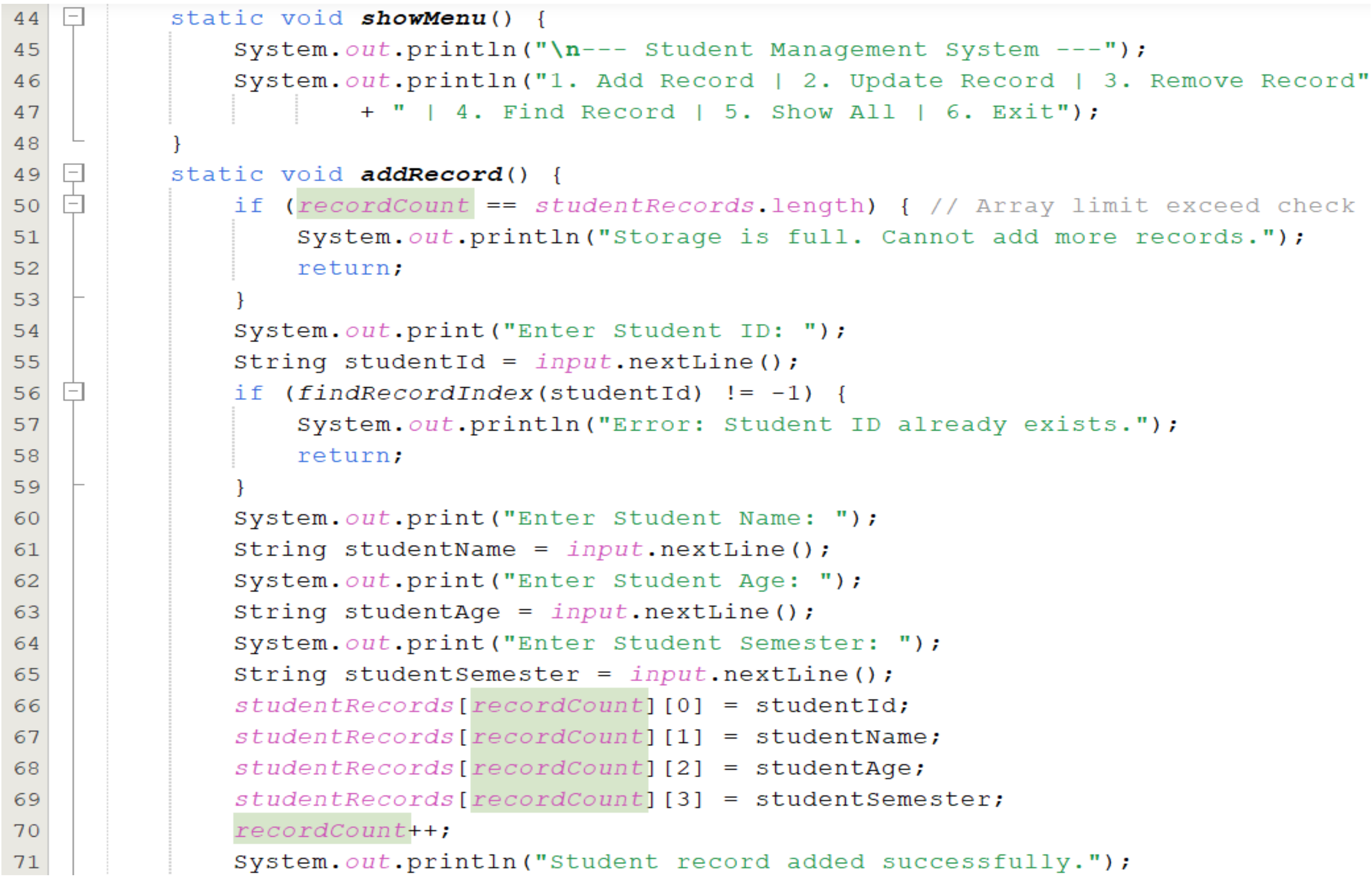
Code:



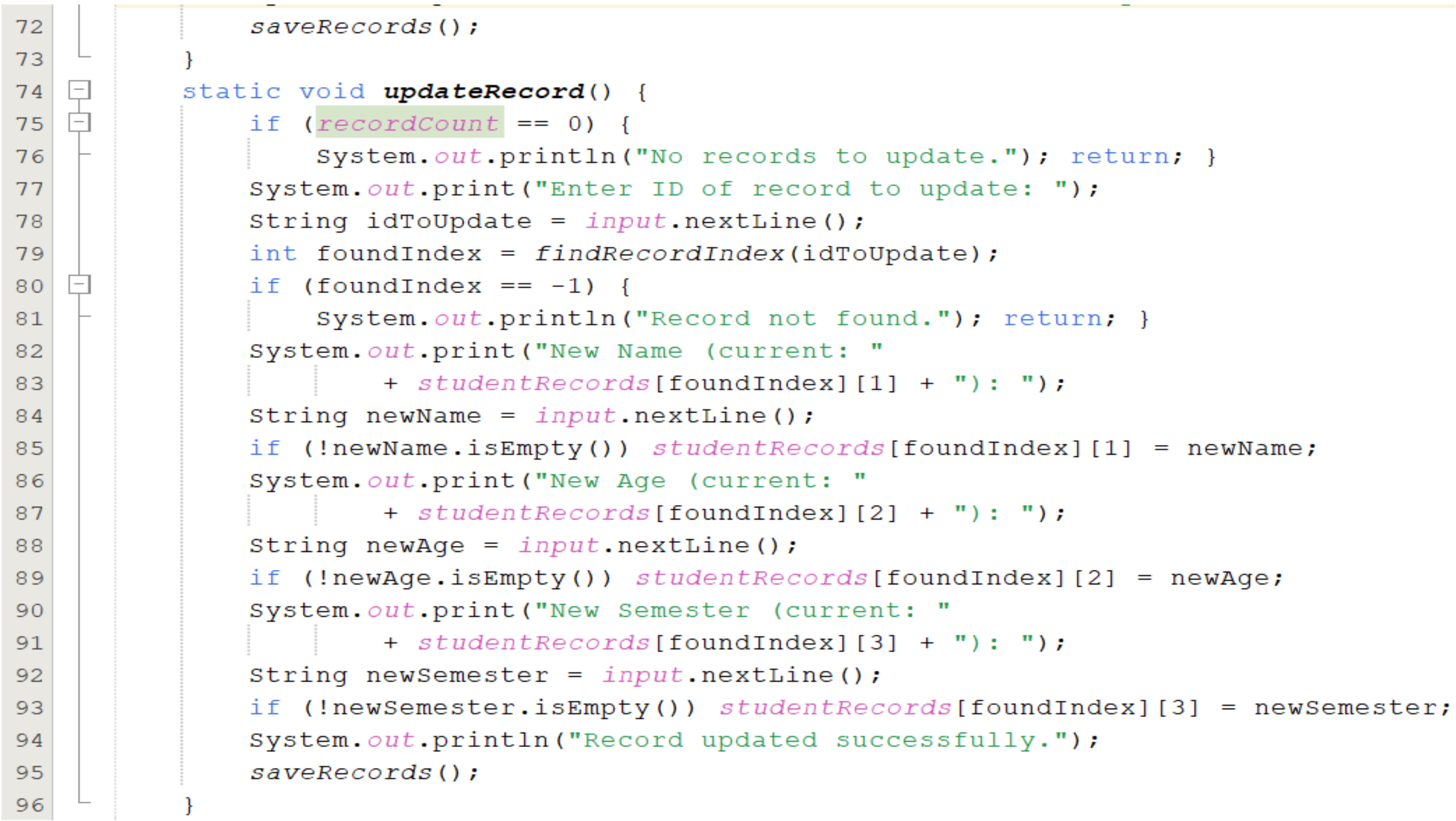
```java
package com.mycompany.enhancedstudentmanager;
import java.util.Scanner;
import java.io.FileWriter;
import java.io.FileReader;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.util.InputMismatchException;

public class EnhancedStudentManager {

    static String[][] studentRecords = new String[5][4];
    static int recordCount = 0;
    static Scanner input = new Scanner(System.in);
    static final String FILENAME = "C:\\Users\\COMPUTER ARENA"
                    + "\\Desktop\\FileHandling\\records.txt";
    public static void main(String[] args) {
        loadRecords();
        int userChoice;
        do {
            showMenu();
            System.out.print("Enter your choice (1-6): ");
            try {
                userChoice = input.nextInt();
            } catch (InputMismatchException e) {
                System.out.println("Invalid input. Please enter a number.");
                userChoice = 0;
```



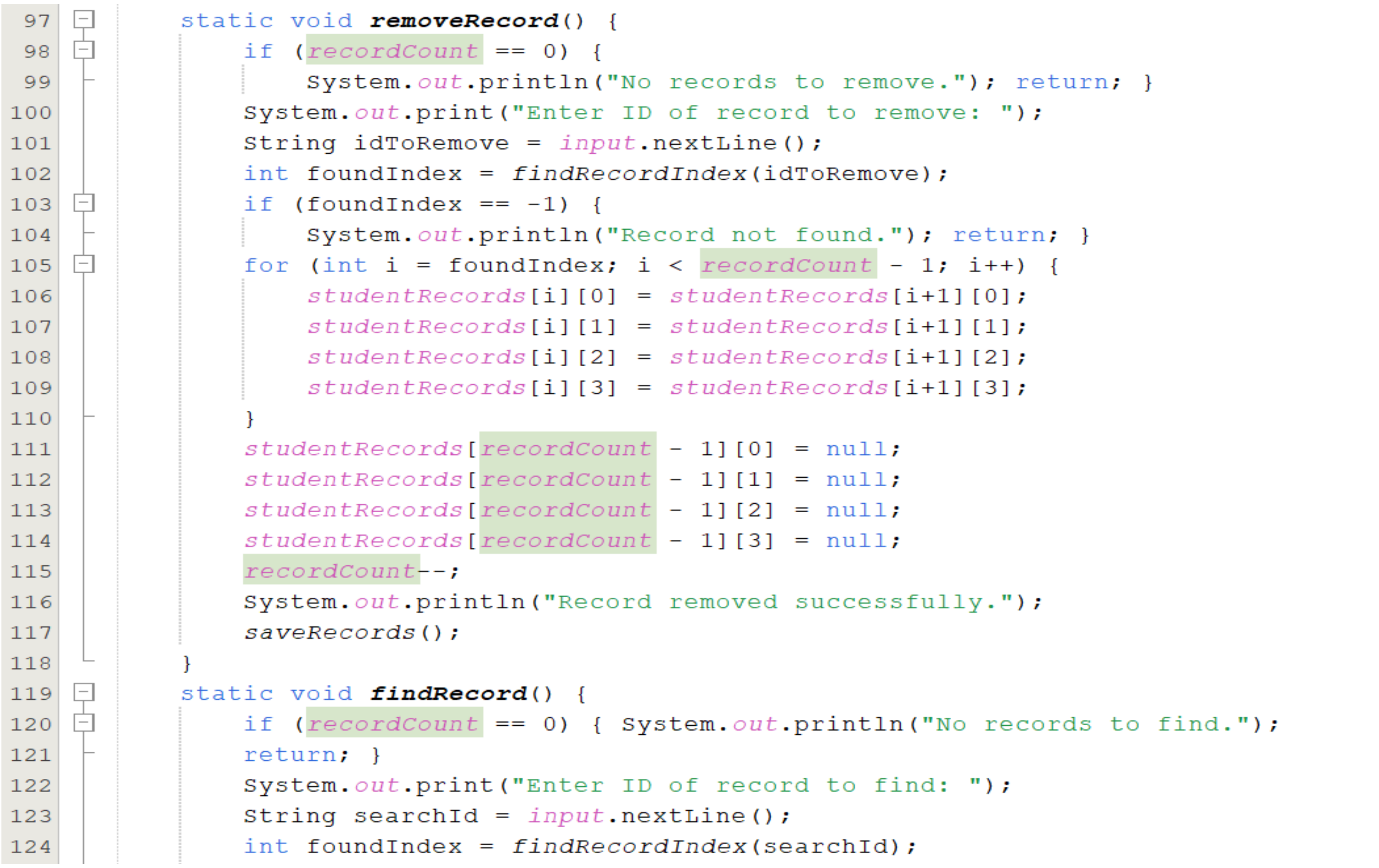
```java
            }
            input.nextLine();
            switch (userChoice) {
                case 1: addRecord(); break;
                case 2: updateRecord(); break;
                case 3: removeRecord(); break;
                case 4: findRecord(); break;
                case 5: showAllRecords(); break;
                case 6: System.out.println("Exiting program. Goodbye!"); break;
                default: System.out.println("Invalid choice.Please try again.");
            }
            System.out.println("---");
        } while (userChoice != 6);
        input.close();
    }
```



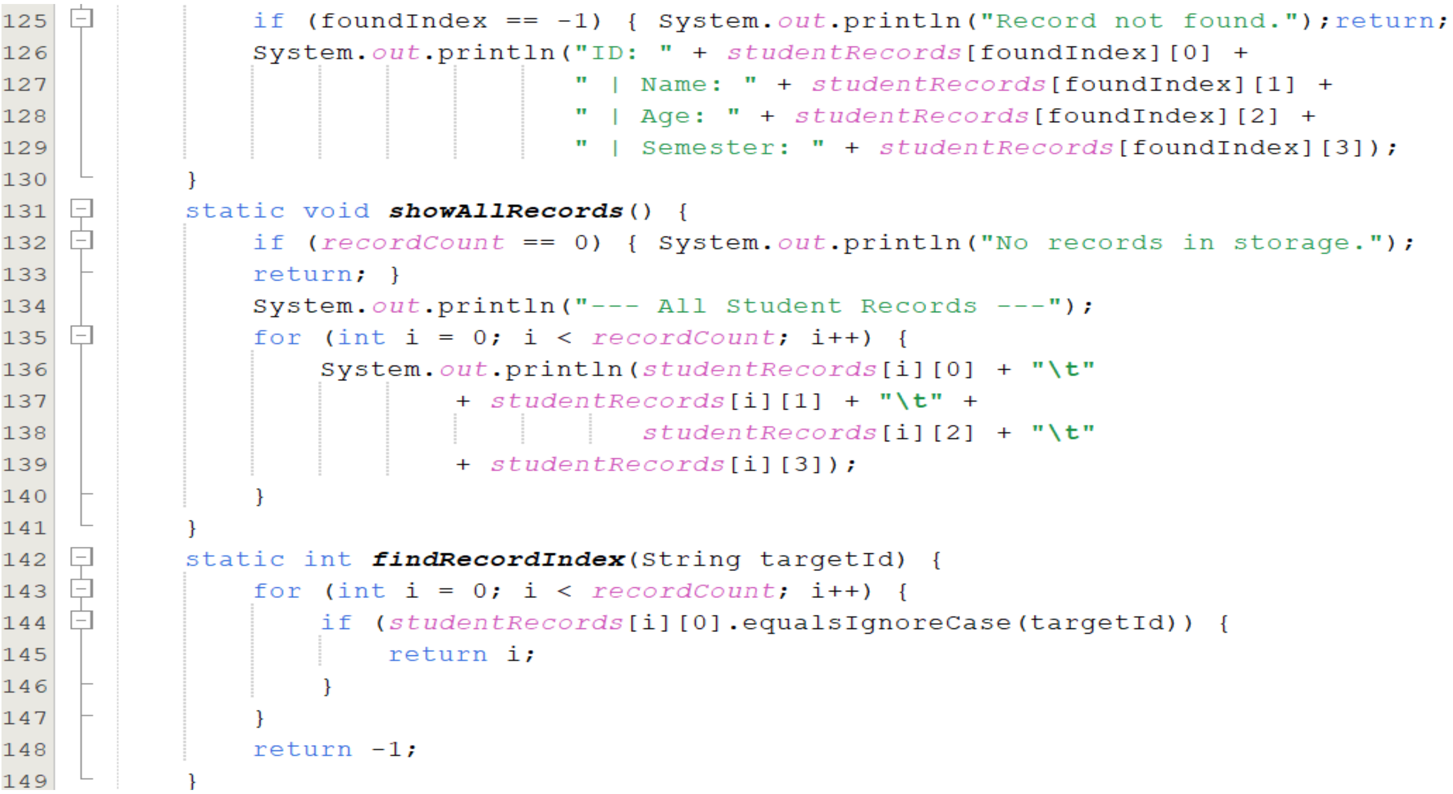
```java
    static void showMenu() {
        System.out.println("\n--- Student Management System ---");
        System.out.println("1. Add Record | 2. Update Record | 3. Remove Record"
                + " | 4. Find Record | 5. Show All | 6. Exit");
    }
    static void addRecord() {
        if (recordCount == studentRecords.length) { // Array limit exceed check
            System.out.println("Storage is full. Cannot add more records.");
            return;
        }
        System.out.print("Enter Student ID: ");
        String studentId = input.nextLine();
        if (findRecordIndex(studentId) != -1) {
            System.out.println("Error: Student ID already exists.");
            return;
        }
        System.out.print("Enter Student Name: ");
        String studentName = input.nextLine();
        System.out.print("Enter Student Age: ");
        String studentAge = input.nextLine();
        System.out.print("Enter Student Semester: ");
        String studentSemester = input.nextLine();
        studentRecords[recordCount][0] = studentId;
        studentRecords[recordCount][1] = studentName;
        studentRecords[recordCount][2] = studentAge;
        studentRecords[recordCount][3] = studentSemester;
        recordCount++;
        System.out.println("Student record added successfully.");
```



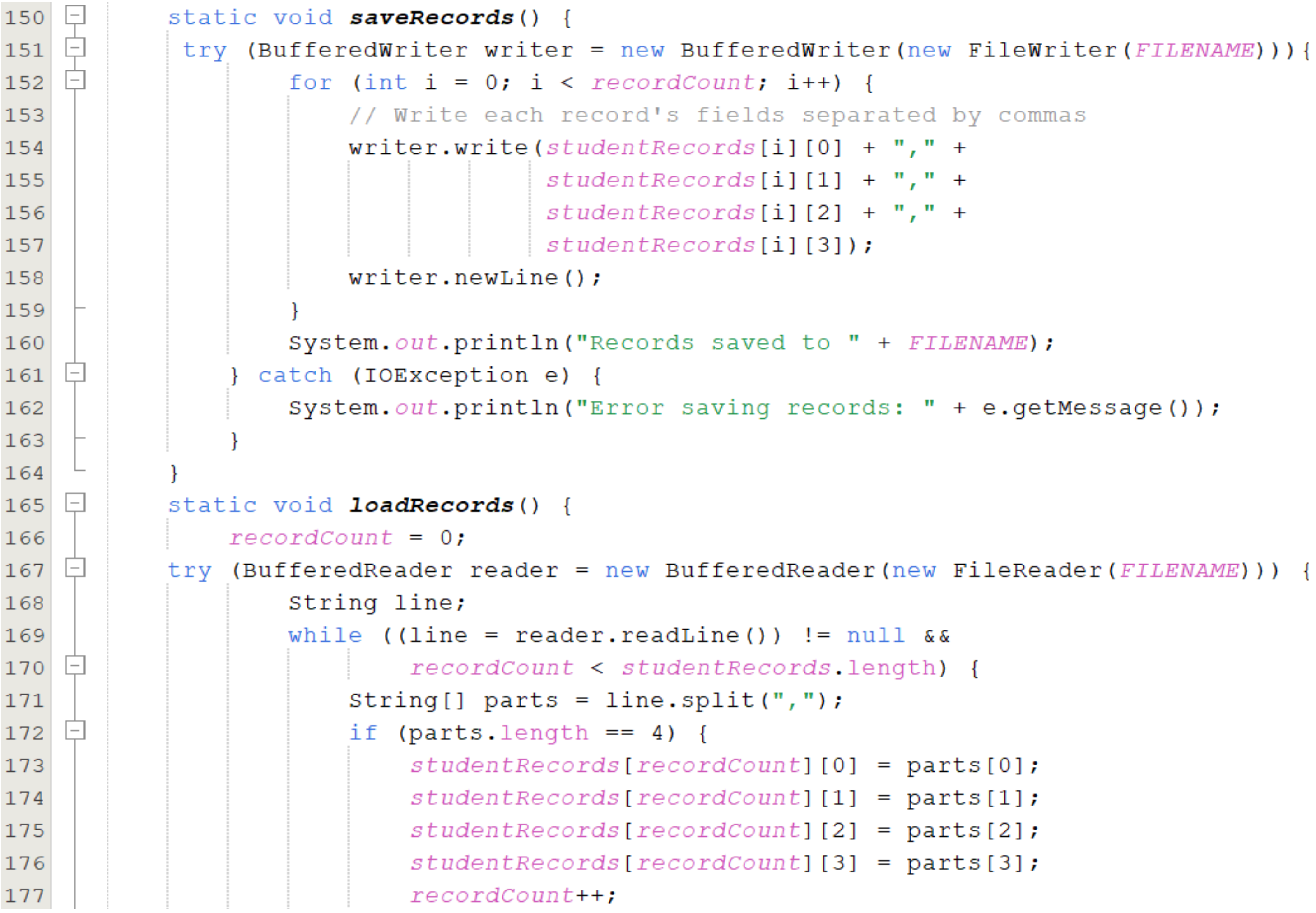
```java
        saveRecords();
    }
    static void updateRecord() {
        if (recordCount == 0) {
            System.out.println("No records to update."); return; }
        System.out.print("Enter ID of record to update: ");
        String idToUpdate = input.nextLine();
        int foundIndex = findRecordIndex(idToUpdate);
        if (foundIndex == -1) {
            System.out.println("Record not found."); return; }
        System.out.print("New Name (current: "
                + studentRecords[foundIndex][1] + "): ");
        String newName = input.nextLine();
        if (!newName.isEmpty()) studentRecords[foundIndex][1] = newName;
        System.out.print("New Age (current: "
                + studentRecords[foundIndex][2] + "): ");
        String newAge = input.nextLine();
        if (!newAge.isEmpty()) studentRecords[foundIndex][2] = newAge;
        System.out.print("New Semester (current: "
                + studentRecords[foundIndex][3] + "): ");
        String newSemester = input.nextLine();
        if (!newSemester.isEmpty()) studentRecords[foundIndex][3] = newSemester;
        System.out.println("Record updated successfully.");
        saveRecords();
    }
```



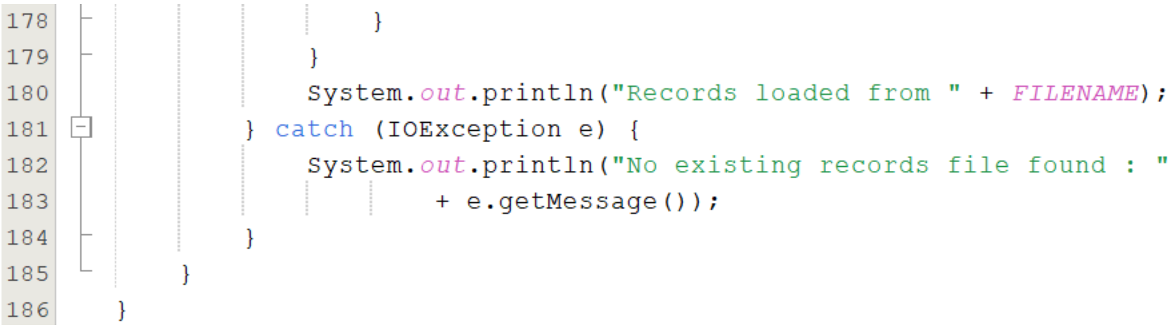
```java
    static void removeRecord() {
        if (recordCount == 0) {
            System.out.println("No records to remove."); return; }
        System.out.print("Enter ID of record to remove: ");
        String idToRemove = input.nextLine();
        int foundIndex = findRecordIndex(idToRemove);
        if (foundIndex == -1) {
            System.out.println("Record not found."); return; }
        for (int i = foundIndex; i < recordCount - 1; i++) {
            studentRecords[i][0] = studentRecords[i+1][0];
            studentRecords[i][1] = studentRecords[i+1][1];
            studentRecords[i][2] = studentRecords[i+1][2];
            studentRecords[i][3] = studentRecords[i+1][3];
        }
        studentRecords[recordCount - 1][0] = null;
        studentRecords[recordCount - 1][1] = null;
        studentRecords[recordCount - 1][2] = null;
        studentRecords[recordCount - 1][3] = null;
        recordCount--;
        System.out.println("Record removed successfully.");
        saveRecords();
    }
    static void findRecord() {
        if (recordCount == 0) { System.out.println("No records to find.");
        return; }
        System.out.print("Enter ID of record to find: ");
        String searchId = input.nextLine();
        int foundIndex = findRecordIndex(searchId);
```



```java
        if (foundIndex == -1) { System.out.println("Record not found.");return;
        System.out.println("ID: " + studentRecords[foundIndex][0] +
                           " | Name: " + studentRecords[foundIndex][1] +
                           " | Age: " + studentRecords[foundIndex][2] +
                           " | Semester: " + studentRecords[foundIndex][3]);
    }
    static void showAllRecords() {
        if (recordCount == 0) { System.out.println("No records in storage.");
        return; }
        System.out.println("--- All Student Records ---");
        for (int i = 0; i < recordCount; i++) {
            System.out.println(studentRecords[i][0] + "\t"
                    + studentRecords[i][1] + "\t" +
                           studentRecords[i][2] + "\t"
                    + studentRecords[i][3]);
        }
    }
    static int findRecordIndex(String targetId) {
        for (int i = 0; i < recordCount; i++) {
            if (studentRecords[i][0].equalsIgnoreCase(targetId)) {
                return i;
            }
        }
        return -1;
    }
```

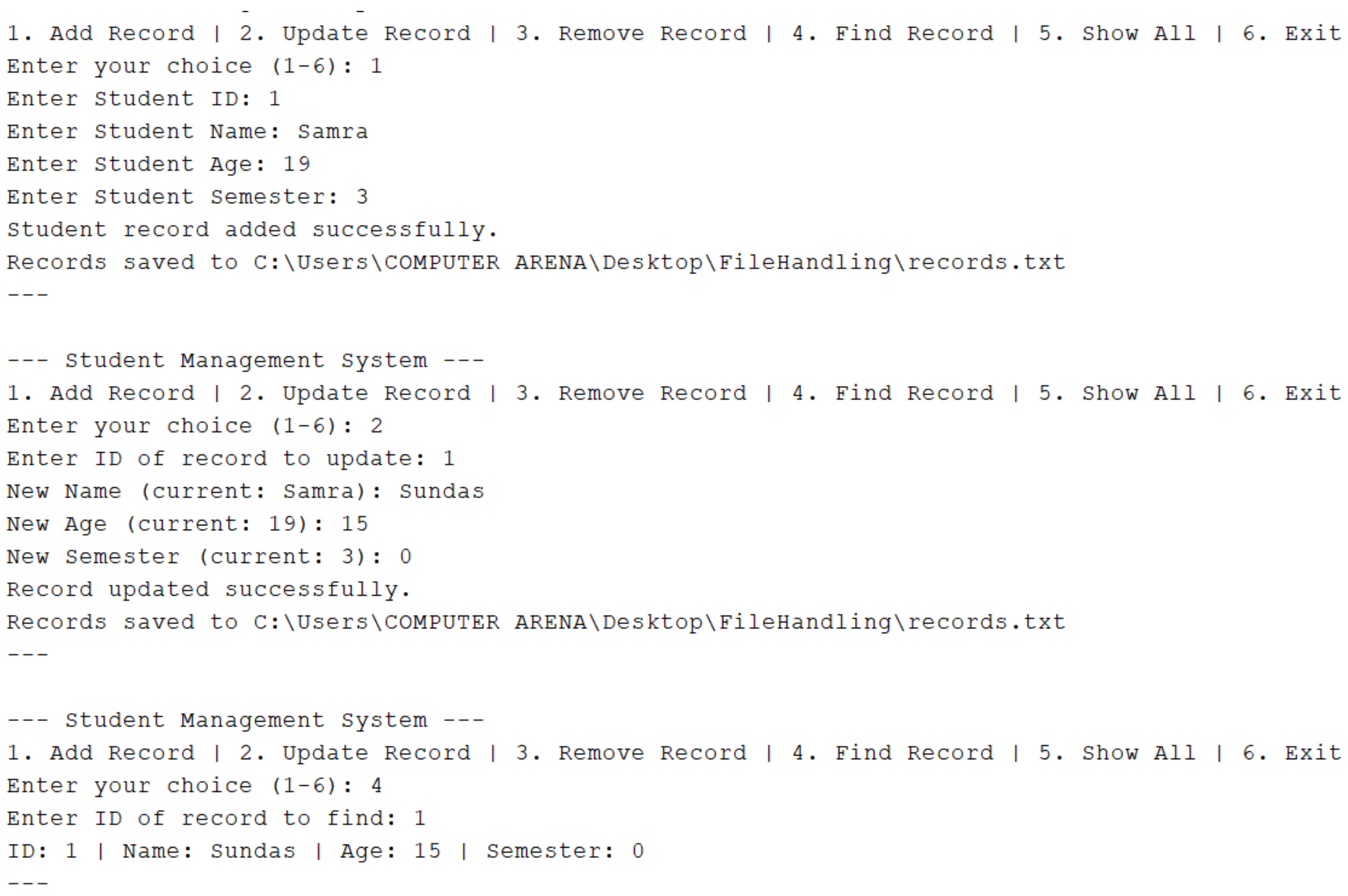


```
    static void saveRecords() {
     try (BufferedWriter writer = new BufferedWriter(new FileWriter(FILENAME))){
            for (int i = 0; i < recordCount; i++) {
                // Write each record's fields separated by commas
                writer.write(studentRecords[i][0] + "," +
                             studentRecords[i][1] + "," +
                             studentRecords[i][2] + "," +
                             studentRecords[i][3]);
                writer.newLine();
            }
            System.out.println("Records saved to " + FILENAME);
        } catch (IOException e) {
            System.out.println("Error saving records: " + e.getMessage());
        }
    }
    static void loadRecords() {
        recordCount = 0;
    try (BufferedReader reader = new BufferedReader(new FileReader(FILENAME))) {
            String line;
            while ((line = reader.readLine()) != null &&
                    recordCount < studentRecords.length) {
                String[] parts = line.split(",");
                if (parts.length == 4) {
                    studentRecords[recordCount][0] = parts[0];
                    studentRecords[recordCount][1] = parts[1];
                    studentRecords[recordCount][2] = parts[2];
                    studentRecords[recordCount][3] = parts[3];
                    recordCount++;
```

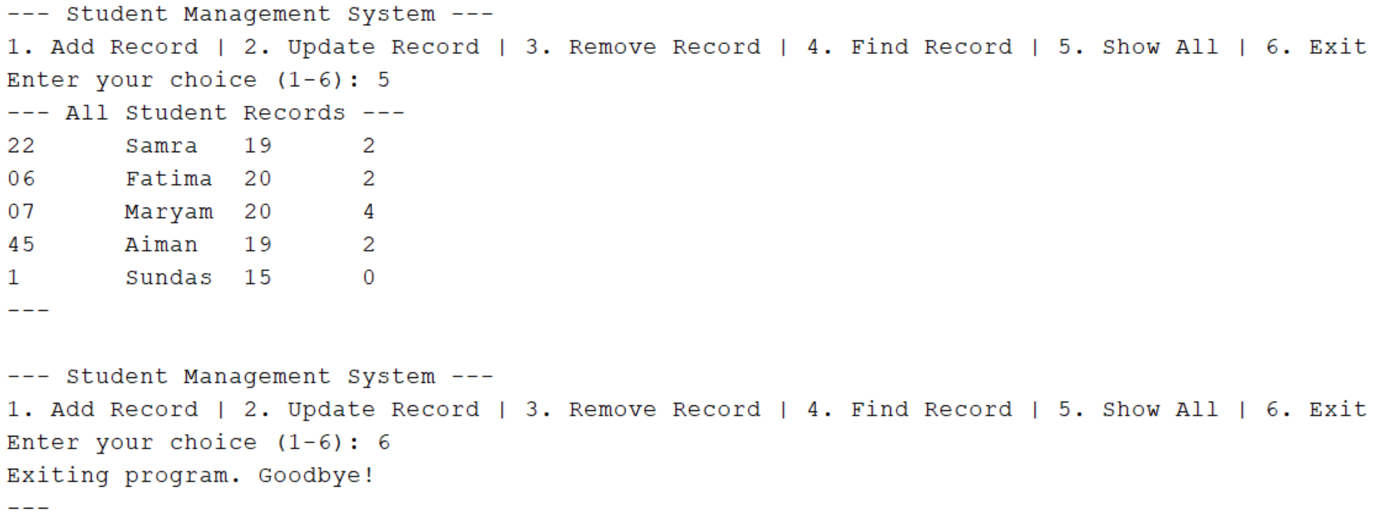


```
                }
            }
            System.out.println("Records loaded from " + FILENAME);
        } catch (IOException e) {
            System.out.println("No existing records file found : "
                    + e.getMessage());
        }
    }
}
```

Output:



```
1. Add Record | 2. Update Record | 3. Remove Record | 4. Find Record | 5. Show All | 6. Exit
Enter your choice (1-6): 1
Enter Student ID: 1
Enter Student Name: Samra
Enter Student Age: 19
Enter Student Semester: 3
Student record added successfully.
Records saved to C:\Users\COMPUTER ARENA\Desktop\FileHandling\records.txt
---

--- Student Management System ---
1. Add Record | 2. Update Record | 3. Remove Record | 4. Find Record | 5. Show All | 6. Exit
Enter your choice (1-6): 2
Enter ID of record to update: 1
New Name (current: Samra): Sundas
New Age (current: 19): 15
New Semester (current: 3): 0
Record updated successfully.
Records saved to C:\Users\COMPUTER ARENA\Desktop\FileHandling\records.txt
---

--- Student Management System ---
1. Add Record | 2. Update Record | 3. Remove Record | 4. Find Record | 5. Show All | 6. Exit
Enter your choice (1-6): 4
Enter ID of record to find: 1
ID: 1 | Name: Sundas | Age: 15 | Semester: 0
---
```



```
--- Student Management System ---
1. Add Record | 2. Update Record | 3. Remove Record | 4. Find Record | 5. Show All | 6. Exit
Enter your choice (1-6): 5
--- All Student Records ---
22      Samra   19      2
06      Fatima  20      2
07      Maryam  20      4
45      Aiman   19      2
1       Sundas  15      0
---

--- Student Management System ---
1. Add Record | 2. Update Record | 3. Remove Record | 4. Find Record | 5. Show All | 6. Exit
Enter your choice (1-6): 6
Exiting program. Goodbye!
---
```