# Student Record Management System - AcademicPortal.java

This program is a simple and efficient Student Record Management System developed in Java. It allows an administrator to register new students, update student information, and view student details using a menu-driven interface. All logic is handled within a single file using static methods and variables, and it uses an array to store up to 100 student records.

## Program Features and Requirements

1. Student Data Storage:

* - Uses private instance variables in a nested class (Learner) to store student name, ID, age, and grade.

2. Student Management:

* - Utilizes private static variables and static methods within a single file structure (AcademicPortal class).
* - Methods include registerLearner, modifyLearner, and viewLearner.

3. Administrator Interface:

* - Menu with options for adding, updating, and viewing student details.
* - Input is handled via Scanner with appropriate validation.

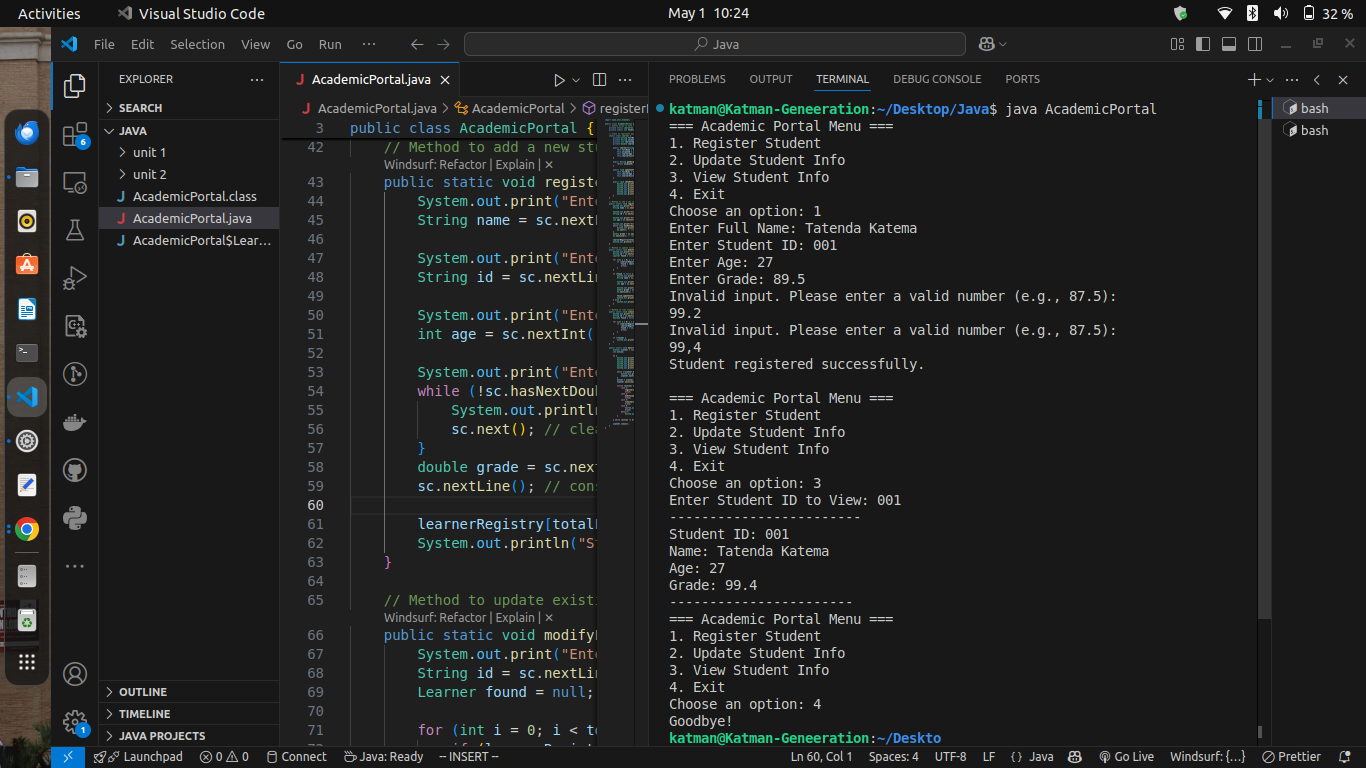
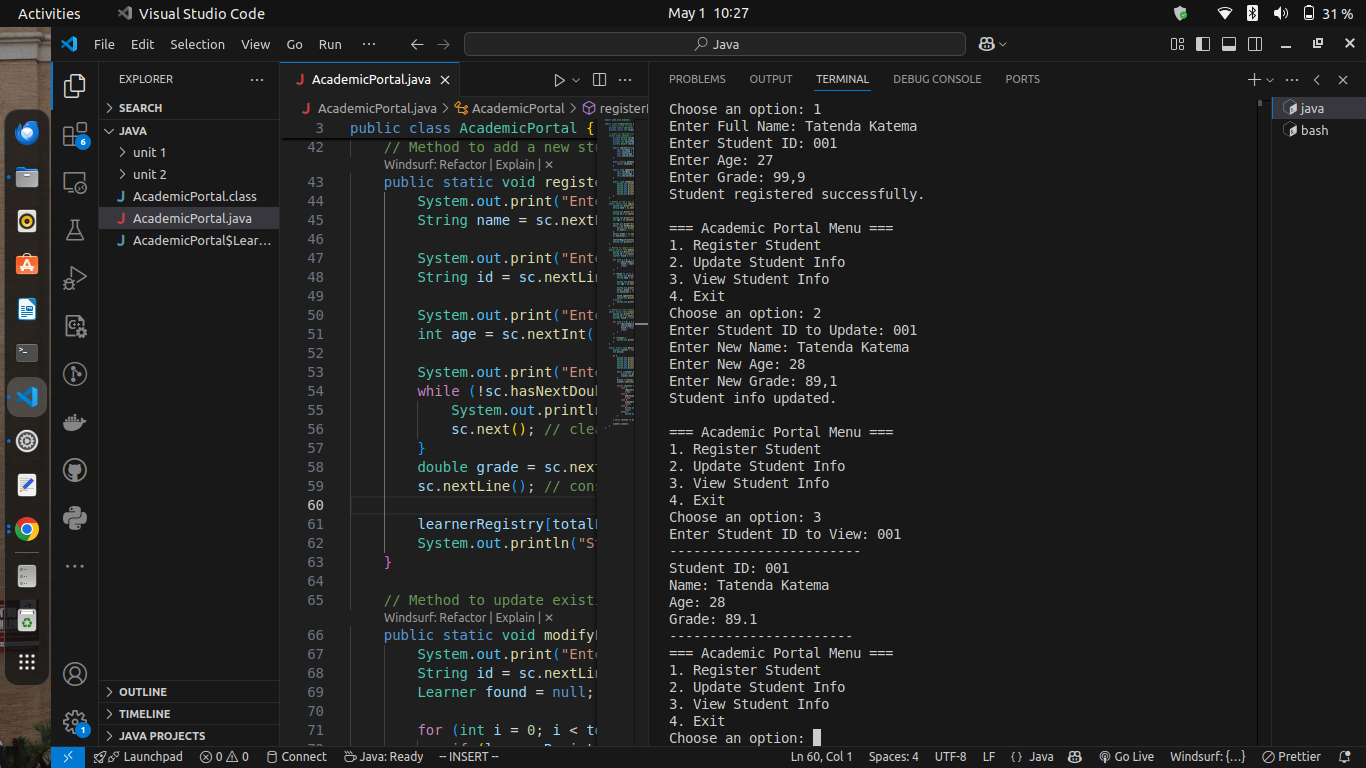
4. Error Handling:

* - Handles invalid ID entries and invalid menu selections.

5. Documentation:

* - Clear and simple comments and structure for readability.

Output:



Code Java:  
  
import java.util.Scanner;

public class AcademicPortal {

// Static array to hold student records

private static Learner[] learnerRegistry = new Learner[100];

private static int totalLearners = 0;

// Class to represent each student

static class Learner {

private String fullName;

private String uniqueId;

private int learnerAge;

private double learnerScore;

public Learner(String fullName, String uniqueId, int learnerAge, double learnerScore) {

this.fullName = fullName;

this.uniqueId = uniqueId;

this.learnerAge = learnerAge;

this.learnerScore = learnerScore;

}

public String getUniqueId() {

return uniqueId;

}

public void updateInfo(String newName, int newAge, double newScore) {

this.fullName = newName;

this.learnerAge = newAge;

this.learnerScore = newScore;

}

public void showDetails() {

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

System.out.println("Student ID: " + uniqueId);

System.out.println("Name: " + fullName);

System.out.println("Age: " + learnerAge);

System.out.println("Grade: " + learnerScore);

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

}

}

// Method to add a new student

public static void registerLearner(Scanner sc) {

System.out.print("Enter Full Name: ");

String name = sc.nextLine();

System.out.print("Enter Student ID: ");

String id = sc.nextLine();

System.out.print("Enter Age: ");

int age = sc.nextInt();

System.out.print("Enter Grade: (use ',' for decimal ) ");

while (!sc.hasNextDouble()) {

System.out.println("Invalid input. Please enter a valid number (e.g., 87,5):");

sc.next(); // clear the invalid input

}

double grade = sc.nextDouble();

sc.nextLine(); // consume newline

learnerRegistry[totalLearners++] = new Learner(name, id, age, grade);

System.out.println("Student registered successfully.\n");

}

// Method to update existing student

public static void modifyLearner(Scanner sc) {

System.out.print("Enter Student ID to Update: ");

String id = sc.nextLine();

Learner found = null;

for (int i = 0; i < totalLearners; i++) {

if (learnerRegistry[i].getUniqueId().equals(id)) {

found = learnerRegistry[i];

break;

}

}

if (found != null) {

System.out.print("Enter New Name: ");

String name = sc.nextLine();

System.out.print("Enter New Age: ");

int age = sc.nextInt();

System.out.print("Enter New Grade: ");

double grade = sc.nextDouble();

sc.nextLine(); // consume newline

found.updateInfo(name, age, grade);

System.out.println("Student info updated.\n");

} else {

System.out.println("Student ID not found.\n");

}

}

// Method to view student details

public static void viewLearner(Scanner sc) {

System.out.print("Enter Student ID to View: ");

String id = sc.nextLine();

boolean found = false;

for (int i = 0; i < totalLearners; i++) {

if (learnerRegistry[i].getUniqueId().equals(id)) {

learnerRegistry[i].showDetails();

found = true;

break;

}

}

if (!found) {

System.out.println("Student ID not found.\n");

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int action;

do {

System.out.println("=== Academic Portal Menu ===");

System.out.println("1. Register Student");

System.out.println("2. Update Student Info");

System.out.println("3. View Student Info");

System.out.println("4. Exit");

System.out.print("Choose an option: ");

while (!scanner.hasNextInt()) {

System.out.print("Invalid input. Enter a number: ");

scanner.next();

}

action = scanner.nextInt();

scanner.nextLine(); // consume newline

switch (action) {

case 1:

registerLearner(scanner);

break;

case 2:

modifyLearner(scanner);

break;

case 3:

viewLearner(scanner);

break;

case 4:

System.out.println("Goodbye!");

break;

default:

System.out.println("Invalid selection.\n");

}

} while (action != 4);

scanner.close();

}

}