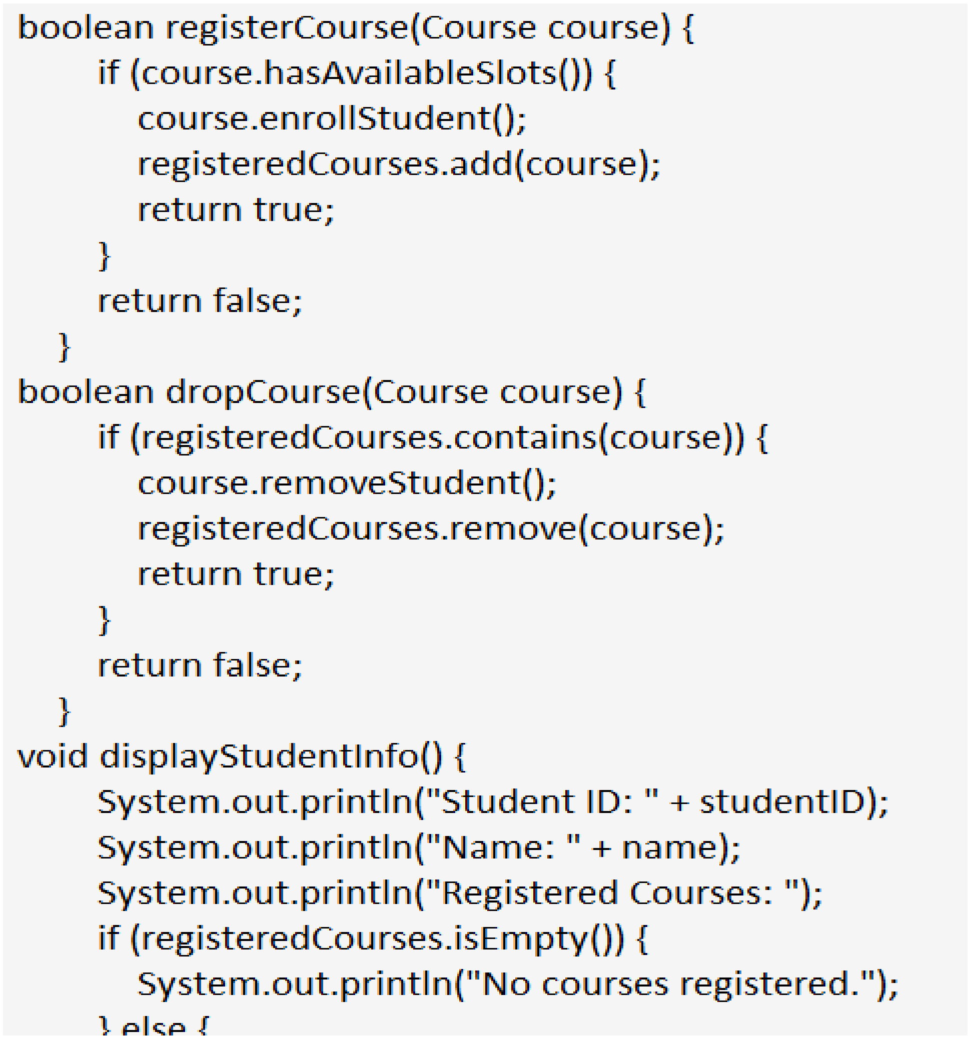
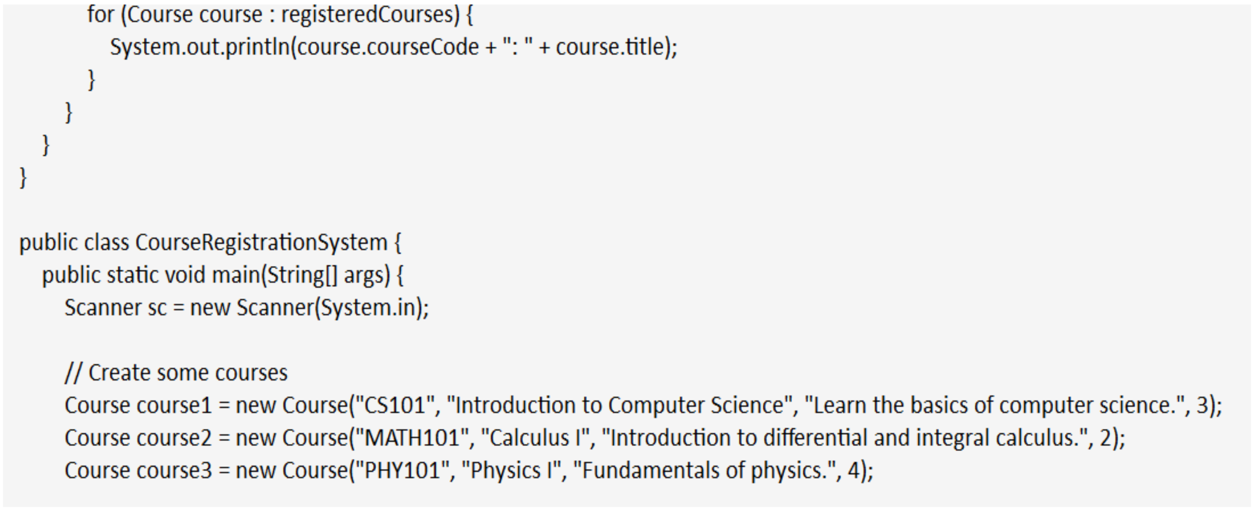
## TASK 5) STUDENT COURSE REGISTRATION SYSTEM

```
import java.util.*;
class Course {
   String courseCode;
   String title;
   String description;
   int capacity;
   int enrolled;
Course(String courseCode, String title, String description, int capacity) {
      this.courseCode = courseCode;
      this.title = title;
      this.description = description;
      this.capacity = capacity;
      this.enrolled = 0;  // initially no students are enrolled
   }
boolean hasAvailableSlots() {
      return enrolled < capacity;
   }
void enrollStudent() {
      if (hasAvailableSlots()) {
         enrolled++;
      }
   }
```

```
void removeStudent() {
        if (enrolled > 0) {
            enrolled--;
        }
    }
void display() {
        System.out.println(courseCode + ": " + title);
        System.out.println("Description: " + description);
        System.out.println("Available Slots: " + (capacity - enrolled));
        System.out.println("Schedule: TBD");
    }
}

class Student {
    String studentID;
    String name;
    List<Course> registeredCourses;

    Student(String studentID, String name) {
        this.studentID = studentID;
        this.name = name;
        this.registeredCourses = new ArrayList<>();
    }
```



```
boolean registerCourse(Course course) {
        if (course.hasAvailableSlots()) {
            course.enrollStudent();
            registeredCourses.add(course);
            return true;
        }
        return false;
    }
boolean dropCourse(Course course) {
        if (registeredCourses.contains(course)) {
            course.removeStudent();
            registeredCourses.remove(course);
            return true;
        }
        return false;
    }
void displayStudentInfo() {
        System.out.println("Student ID: " + studentID);
        System.out.println("Name: " + name);
        System.out.println("Registered Courses: ");
        if (registeredCourses.isEmpty()) {
            System.out.println("No courses registered.");
        } else {
```



```
            for (Course course : registeredCourses) {
                System.out.println(course.courseCode + ": " + course.title);
            }
        }
    }
}

public class CourseRegistrationSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Create some courses
        Course course1 = new Course("CS101", "Introduction to Computer Science", "Learn the basics of computer science.", 3);
        Course course2 = new Course("MATH101", "Calculus I", "Introduction to differential and integral calculus.", 2);
        Course course3 = new Course("PHY101", "Physics I", "Fundamentals of physics.", 4);
```

```java
// Create some students
Student student1 = new Student("S001", "Anuja");
Student student2 = new Student("S002", "Shreya");

// Store courses and students
List<Course> courses = new ArrayList<>();
courses.add(course1);
courses.add(course2);
courses.add(course3);

List<Student> students = new ArrayList<>();
students.add(student1);
students.add(student2);
```

```
        // Store courses and students
        List<Course> courses = new ArrayList<>();
        courses.add(course1);
        courses.add(course2);
        courses.add(course3);

        List<Student> students = new ArrayList<>();
        students.add(student1);
        students.add(student2);

        // Main menu loop
        while (true) {
            System.out.println("\n--- Course Registration System ---");
            System.out.println("1. View Available Courses");
            System.out.println("2. Register for a Course");
            System.out.println("3. Drop a Course");
            System.out.println("4. View Student Information");
            System.out.println("5. Exit");
            System.out.print("Choose an option: ");
            int choice = sc.nextInt();
            sc.nextLine();  // Consume newline
```

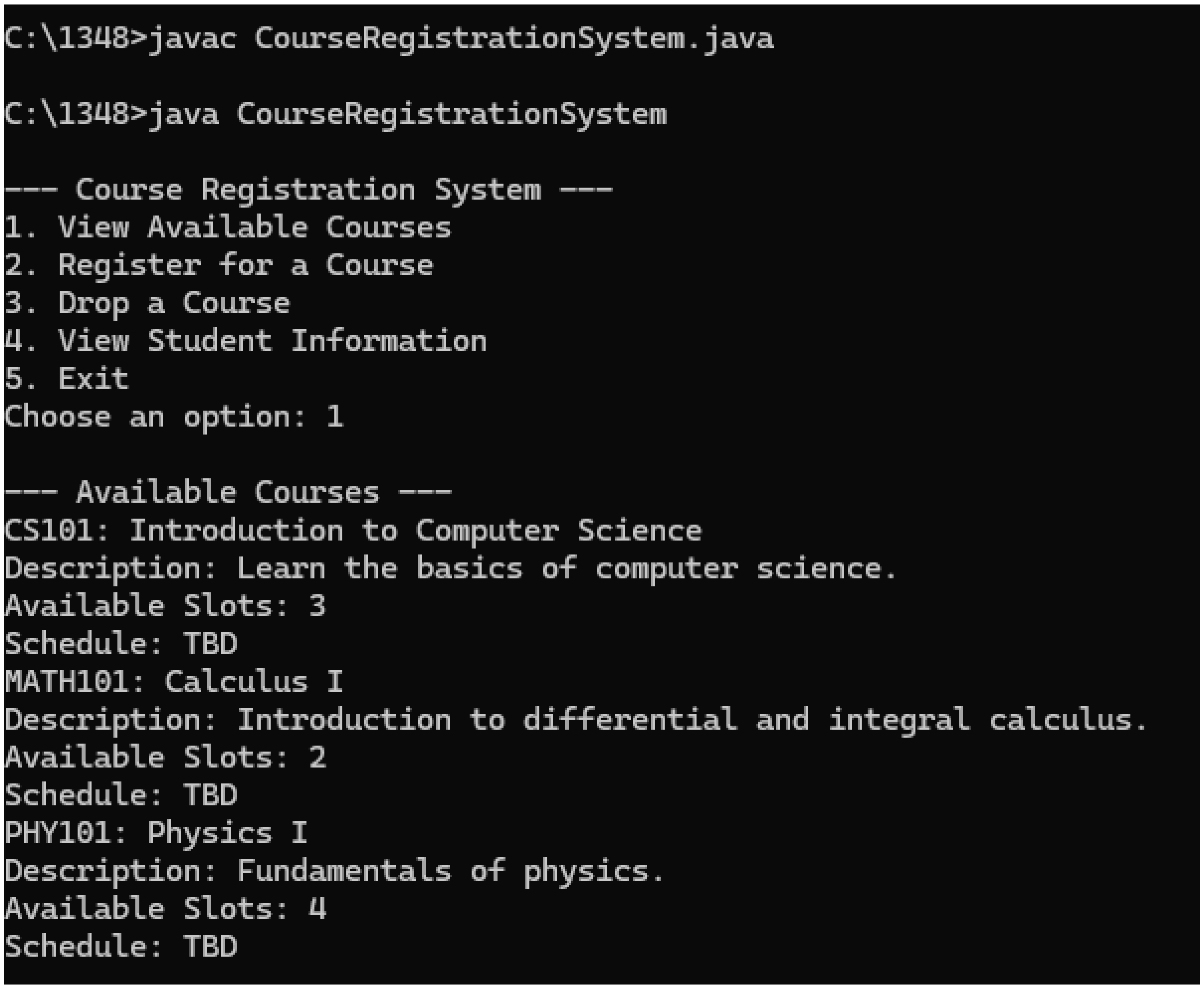
```
switch (choice) {
  case 1:
    System.out.println("\n--- Available Courses ---");
    for (Course course : courses) {
      course.display();
    }
    break;

  case 2:
    System.out.print("Enter your Student ID: ");
    String studentID = sc.nextLine();
    Student student = findStudentByID(students, studentID);
    if (student != null) {
      System.out.print("Enter Course Code to register: ");
      String courseCode = sc.nextLine();
      Course courseToRegister = findCourseByCode(courses, courseCode);
      if (courseToRegister != null && student.registerCourse(courseToRegister)) {
        System.out.println("Successfully registered for " + courseToRegister.title);
      } else {
        System.out.println("Failed to register. Either the course is full or invalid course code.");
      }
```

```
      }
    } else {
      System.out.println("Student not found.");
    }
    break;

  case 3:
    System.out.print("Enter your Student ID: ");
    studentID = sc.nextLine();
    student = findStudentByID(students, studentID);
    if (student != null) {
      System.out.print("Enter Course Code to drop: ");
      CourseCode = sc.nextLine();
      Course courseToDrop = findCourseByCode(courses, CourseCode);
      if (courseToDrop != null && student.dropCourse(courseToDrop)) {
        System.out.println("Successfully dropped " + courseToDrop.title);
      } else {
        System.out.println("Failed to drop course. Either you are not registered or invalid course code.");
      }
    } else {
      System.out.println("Student not found.");
    }
    break;
```

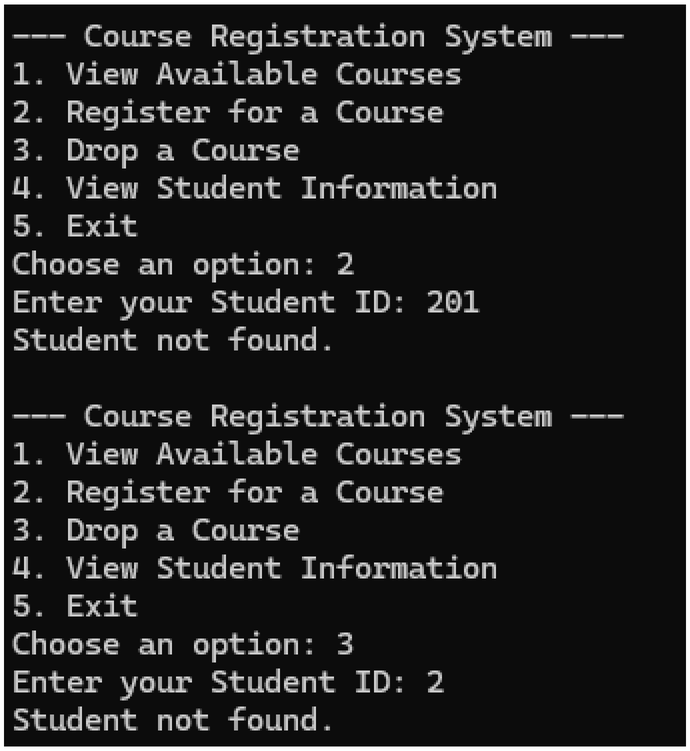
```
            case 4:
                System.out.print("Enter your Student ID: ");
                studentID = sc.nextLine();
                student = findStudentByID(students, studentID);
                if (student != null) {
                    student.displayStudentInfo();
                } else {
                    System.out.println("Student not found.");
                }
                break;

            case 5:
                System.out.println("Exiting the system.");
                sc.close();
                return;

            default:
                System.out.println("Invalid option. Please try again.");
                break;
        }
    }
}
```

```
    // Helper function to find a student by ID
    private static Student findStudentByID(List<Student> students, String studentID) {
        for (Student student : students) {
            if (student.studentID.equals(studentID)) {
                return student;
            }
        }
        return null;
    }

    // Helper function to find a course by course code
    private static Course findCourseByCode(List<Course> courses, String courseCode) {
        for (Course course : courses) {
            if (course.courseCode.equals(courseCode)) {
                return course;
            }
        }
        return null;
    }
}
```

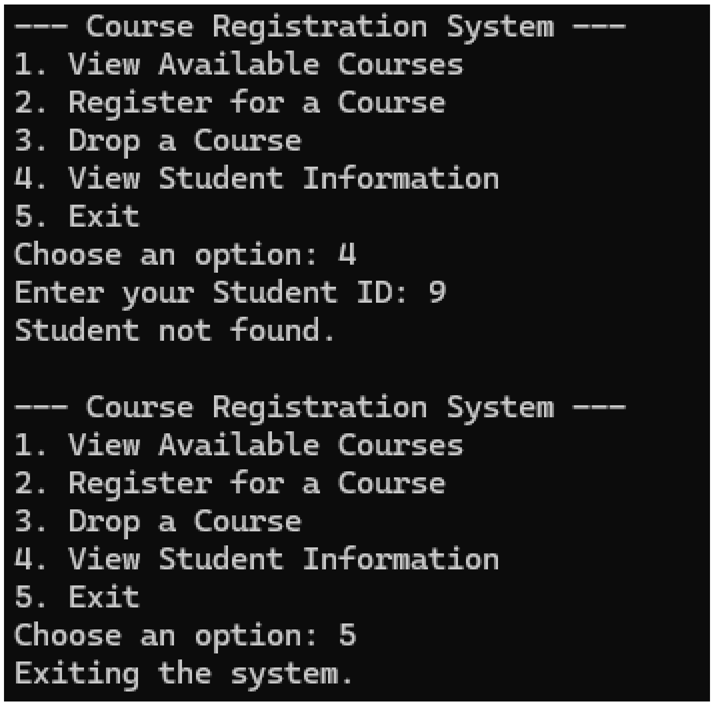
**OUTPUT:-**



```
C:\1348>javac CourseRegistrationSystem.java

C:\1348>java CourseRegistrationSystem

--- Course Registration System ---
1. View Available Courses
2. Register for a Course
3. Drop a Course
4. View Student Information
5. Exit
Choose an option: 1

--- Available Courses ---
CS101: Introduction to Computer Science
Description: Learn the basics of computer science.
Available Slots: 3
Schedule: TBD
MATH101: Calculus I
Description: Introduction to differential and integral calculus.
Available Slots: 2
Schedule: TBD
PHY101: Physics I
Description: Fundamentals of physics.
Available Slots: 4
Schedule: TBD
```



```
--- Course Registration System ---
1. View Available Courses
2. Register for a Course
3. Drop a Course
4. View Student Information
5. Exit
Choose an option: 2
Enter your Student ID: 201
Student not found.

--- Course Registration System ---
1. View Available Courses
2. Register for a Course
3. Drop a Course
4. View Student Information
5. Exit
Choose an option: 3
Enter your Student ID: 2
Student not found.
```



```
--- Course Registration System ---
1. View Available Courses
2. Register for a Course
3. Drop a Course
4. View Student Information
5. Exit
Choose an option: 4
Enter your Student ID: 9
Student not found.

--- Course Registration System ---
1. View Available Courses
2. Register for a Course
3. Drop a Course
4. View Student Information
5. Exit
Choose an option: 5
Exiting the system.
```