Anuja Shiravale
Batch - APRIL BATCH B22

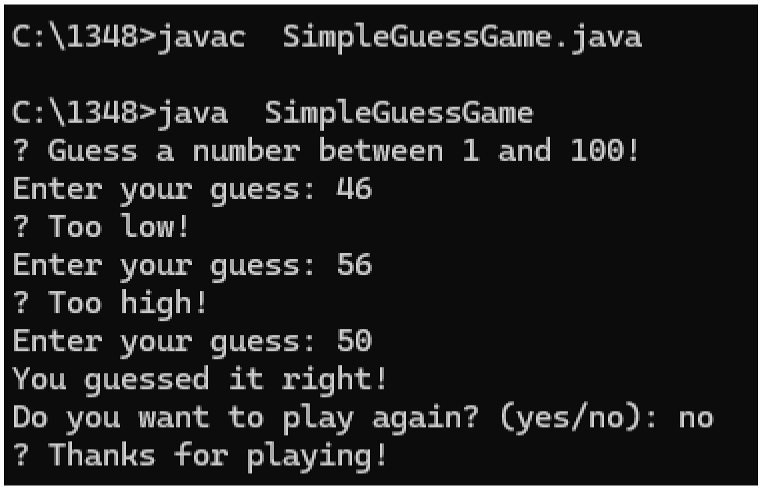
TASK 1) NUMBER GAME

```
import java.util.Scanner;
import java.util.Random;
public class SimpleGuessGame {
public static void main(String[] args) {
Scanner input = new Scanner(System.in);
Random rand = new Random();
int maxAttempts = 5;
String playAgain;
do {
int number = rand.nextInt(100) + 1;
int attempts = 0;
boolean correct = false;
System.out.println("🎯 Guess a number between 1 and 100!");
while (attempts < maxAttempts) {
System.out.print("Enter your guess: ");
int guess = input.nextInt();
attempts++;
if (guess == number) {
System.out.println("You guessed it right!");
correct = true;
break;
} else if (guess < number) {
System.out.println("📉 Too low!");
 } else {
System.out.println("📈 Too high!");
 }
 }
```

Anuja Shiravale
Batch - APRIL BATCH B22

```
if (!correct) {
System.out.println(" Out of attempts! The number was: " + number);
}
System.out.print("Do you want to play again? (yes/no): ");
playAgain = input.next().toLowerCase();
} while (playAgain.equals("yes"));
System.out.println("👋 Thanks for playing!");
input.close();
}
}
```

OUTPUT :-



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

C:\1348>java  SimpleGuessGame
? Guess a number between 1 and 100!
Enter your guess: 46
? Too low!
Enter your guess: 56
? Too high!
Enter your guess: 50
You guessed it right!
Do you want to play again? (yes/no): no
? Thanks for playing!
```

Anuja Shiravale
Batch - APRIL BATCH B22

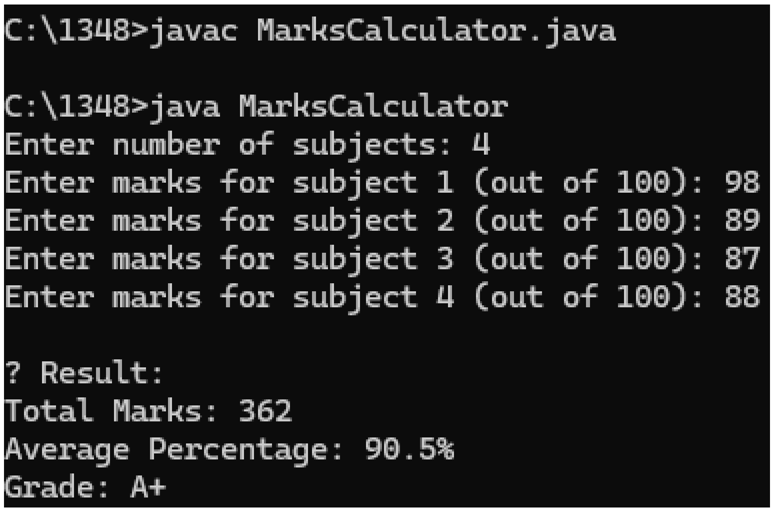
<u>TASK 2) STUDENT GRADE CALCULATOR</u>

```
import java.util.Scanner;
public class MarksCalculator {
public static void main(String[] args) {
Scanner input = new Scanner(System.in);
System.out.print("Enter number of subjects: ");
int numSubjects = input.nextInt();
int totalMarks = 0;
for (int i = 1; i <= numSubjects; i++) {
System.out.print("Enter marks for subject " + i + " (out of 100): ");
int marks = input.nextInt()
while (marks < 0 || marks > 100) {
System.out.print("Invalid marks. Enter again for subject " + i + ": ");
marks = input.nextInt();
}
totalMarks += marks;
}
double average = (double) totalMarks / numSubjects;
String grade;
if (average >= 90) {
grade = "A+";
} else if (average >= 80) {
grade = "A";
} else if (average >= 70) {
grade = "B";
} else if (average >= 60) {
grade = "C";
} else if (average >= 50) {
grade = "D";
} else {
grade = "F";
}
```

Anuja Shiravale
Batch - APRIL BATCH B22

```
System.out.println("\n📊 Result:");
System.out.println("Total Marks: " + totalMarks);
System.out.println("Average Percentage: " + average + "%");
System.out.println("Grade: " + grade);
input.close();
}
}
```

OUTPUT:-



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

C:\1348>java MarksCalculator
Enter number of subjects: 4
Enter marks for subject 1 (out of 100): 98
Enter marks for subject 2 (out of 100): 89
Enter marks for subject 3 (out of 100): 87
Enter marks for subject 4 (out of 100): 88

? Result:
Total Marks: 362
Average Percentage: 90.5%
Grade: A+
```

Anuja Shiravale
Batch - APRIL BATCH B22

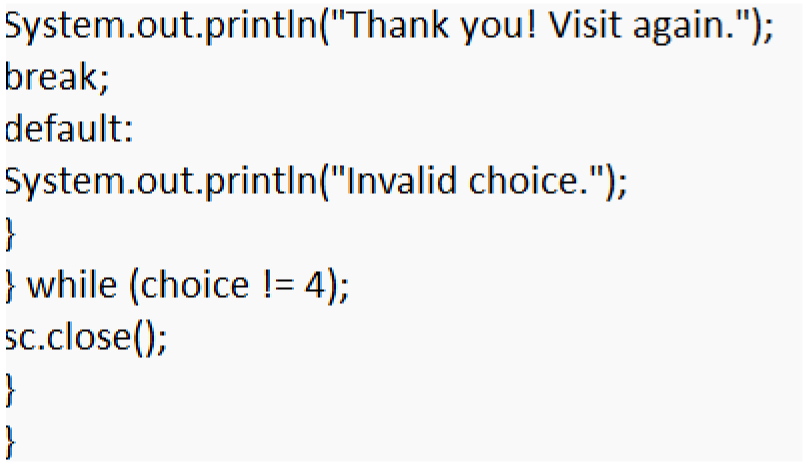
TASK 3) ATM INTERFACE

```
import java.util.Scanner;
class BankAccount {
double balance;
BankAccount(double balance) {
this.balance = balance;
}
void deposit(double amount) {
if (amount > 0) {
balance += amount;
System.out.println("Money deposited successfully.");
} else {
System.out.println("Enter a valid amount.");
}
}
void withdraw(double amount) {
if (amount > 0 && amount <= balance) {
balance -= amount;
System.out.println("Money withdrawn successfully.");
} else {
System.out.println("Insufficient balance or invalid amount.");
}
}
```

Anuja Shiravale
Batch - APRIL BATCH B22

```
void checkBalance() {
System.out.println("Current Balance: ₹" + balance);
}
}
public class ATM {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
BankAccount account = new BankAccount(1000);
int choice;
do {
System.out.println("\n--- ATM Menu ---");
System.out.println("1. Check Balance");
System.out.println("2. Deposit Money");
System.out.println("3. Withdraw Money");
System.out.println("4. Exit");
System.out.print("Enter your choice: ");
choice = sc.nextInt();
switch (choice) {
case 1:
account.checkBalance();
break;
case 2:
System.out.print("Enter amount to deposit: ₹");
double depositAmount = sc.nextDouble();
account.deposit(depositAmount);
break;
case 3:
System.out.print("Enter amount to withdraw: ₹");
double withdrawAmount = sc.nextDouble();
account.withdraw(withdrawAmount);
break;
case 4:
```

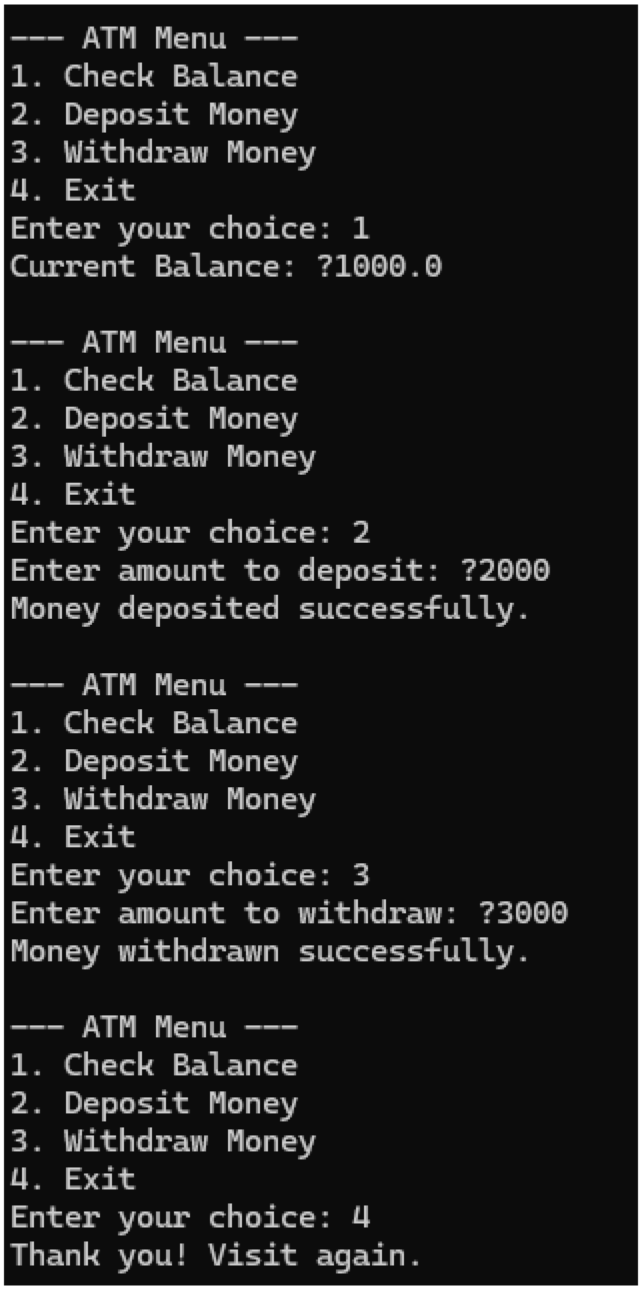
Anuja Shiravale
Batch - APRIL BATCH B22



```
System.out.println("Thank you! Visit again.");
break;
default:
System.out.println("Invalid choice.");
}
} while (choice != 4);
sc.close();
}
}
```

Anuja Shiravale
Batch - APRIL BATCH B22

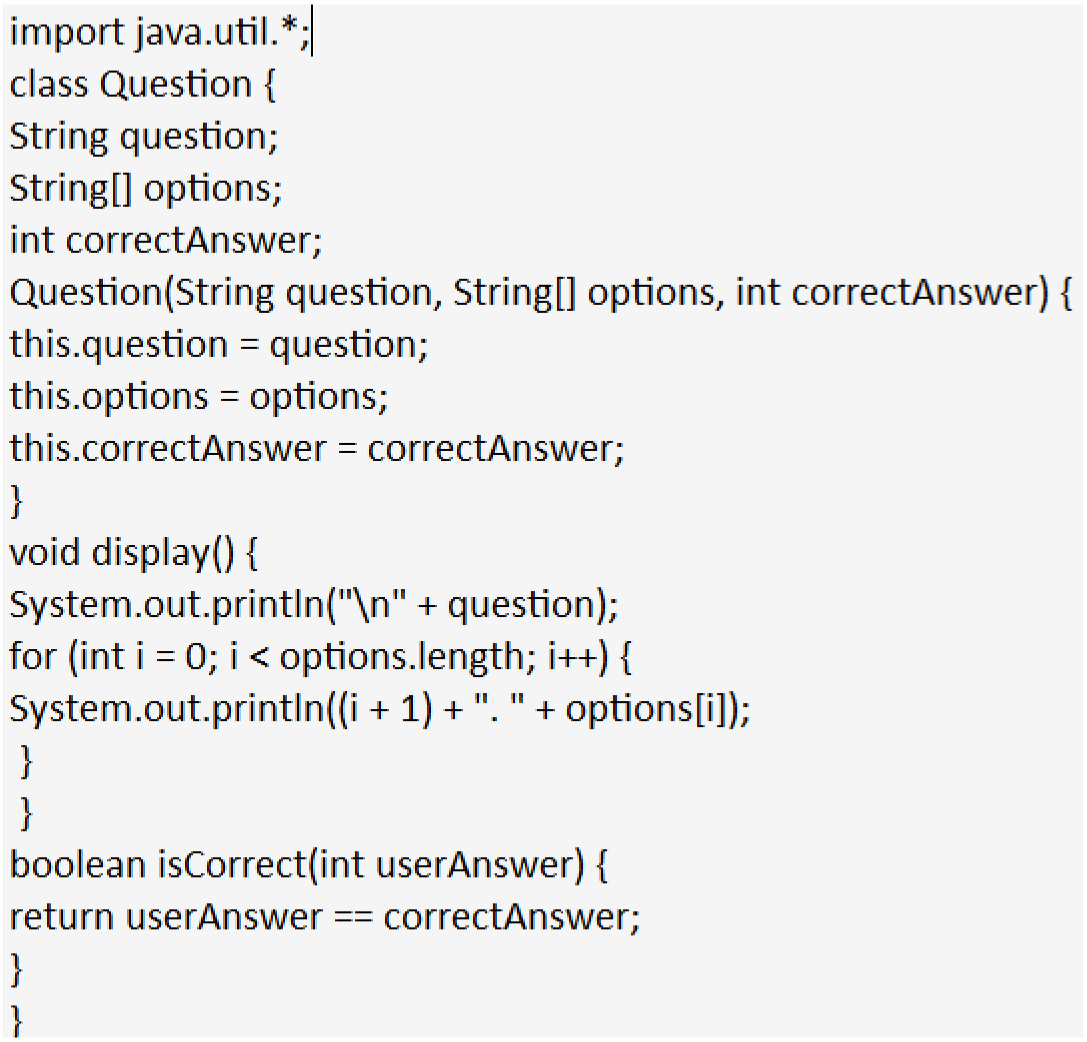
OUTPUT:-



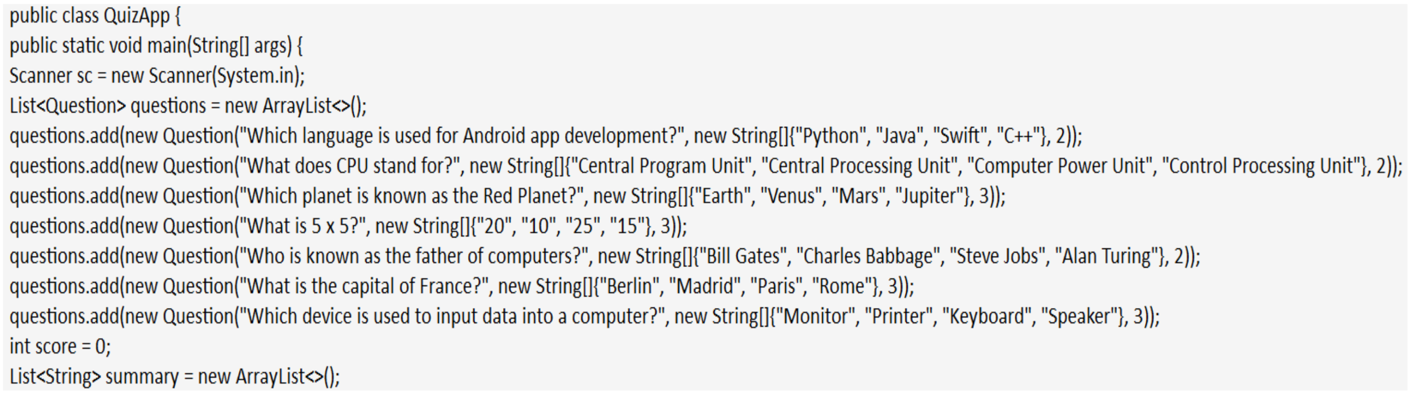
```
--- ATM Menu ---
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 1
Current Balance: ?1000.0

--- ATM Menu ---
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 2
Enter amount to deposit: ?2000
Money deposited successfully.

--- ATM Menu ---
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 3
Enter amount to withdraw: ?3000
Money withdrawn successfully.

--- ATM Menu ---
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 4
Thank you! Visit again.
```

Anuja Shiravale
Batch - APRIL BATCH B22

TASK 4) QUIZ APPLICATION WITH TIMER



```
import java.util.*;
class Question {
String question;
String[] options;
int correctAnswer;
Question(String question, String[] options, int correctAnswer) {
this.question = question;
this.options = options;
this.correctAnswer = correctAnswer;
}
void display() {
System.out.println("\n" + question);
for (int i = 0; i < options.length; i++) {
System.out.println((i + 1) + ". " + options[i]);
 }
 }
boolean isCorrect(int userAnswer) {
return userAnswer == correctAnswer;
}
}
```



```
public class QuizApp {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
List<Question> questions = new ArrayList<>();
questions.add(new Question("Which language is used for Android app development?", new String[]{"Python", "Java", "Swift", "C++"}, 2));
questions.add(new Question("What does CPU stand for?", new String[]{"Central Program Unit", "Central Processing Unit", "Computer Power Unit", "Control Processing Unit"}, 2));
questions.add(new Question("Which planet is known as the Red Planet?", new String[]{"Earth", "Venus", "Mars", "Jupiter"}, 3));
questions.add(new Question("What is 5 x 5?", new String[]{"20", "10", "25", "15"}, 3));
questions.add(new Question("Who is known as the father of computers?", new String[]{"Bill Gates", "Charles Babbage", "Steve Jobs", "Alan Turing"}, 2));
questions.add(new Question("What is the capital of France?", new String[]{"Berlin", "Madrid", "Paris", "Rome"}, 3));
questions.add(new Question("Which device is used to input data into a computer?", new String[]{"Monitor", "Printer", "Keyboard", "Speaker"}, 3));
int score = 0;
List<String> summary = new ArrayList<>();
```

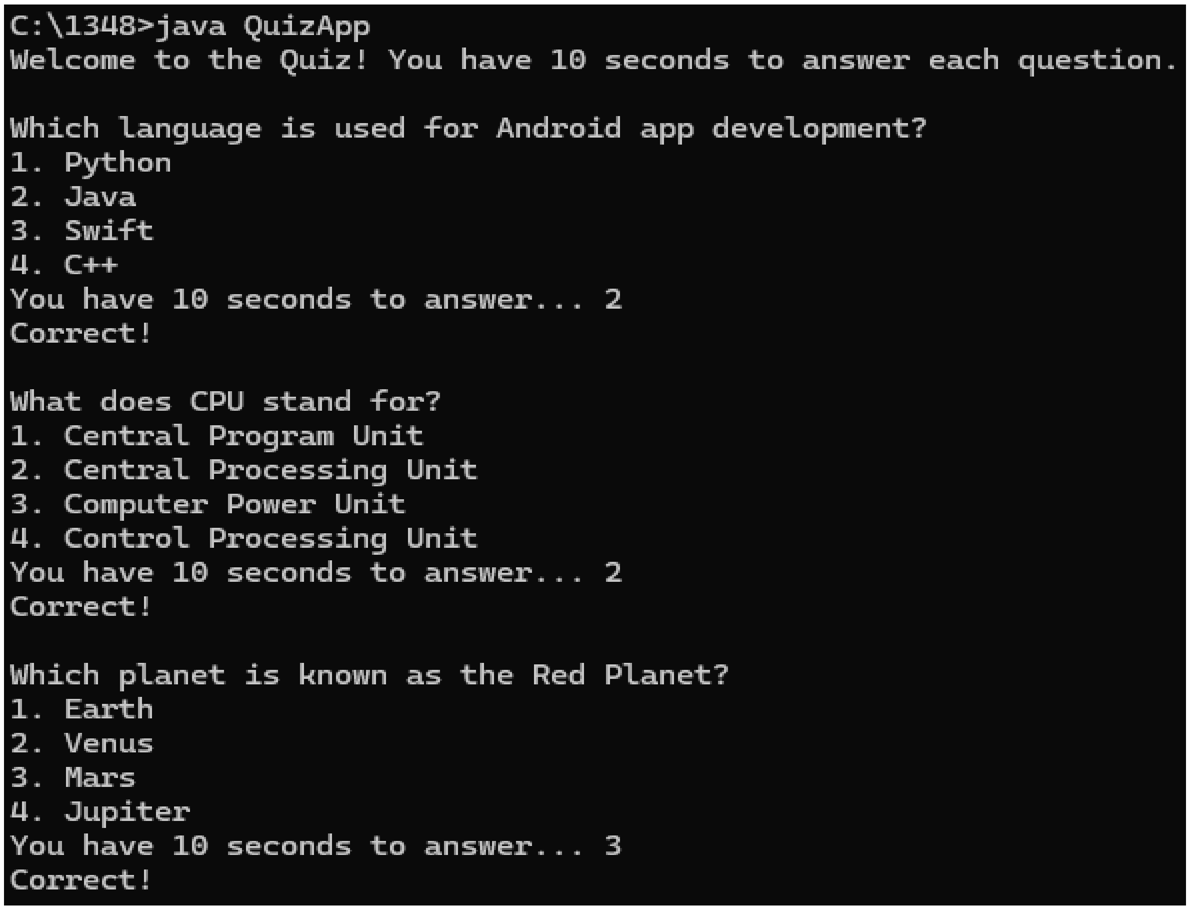
Anuja Shiravale
Batch - APRIL BATCH B22

```
System.out.println("Welcome to the Quiz! You have 10 seconds to answer each question.");
for (int i = 0; i < questions.size(); i++) {
Question q = questions.get(i);
q.display();
int seconds = 10;
System.out.print("You have " + seconds + " seconds to answer... ");
long startTime = System.currentTimeMillis();
int userAnswer = 0;
boolean answered = false;
while ((System.currentTimeMillis() - startTime) / 1000 < seconds) {
if (sc.hasNextInt()) {
userAnswer = sc.nextInt();
answered = true;
break;
}
}
```

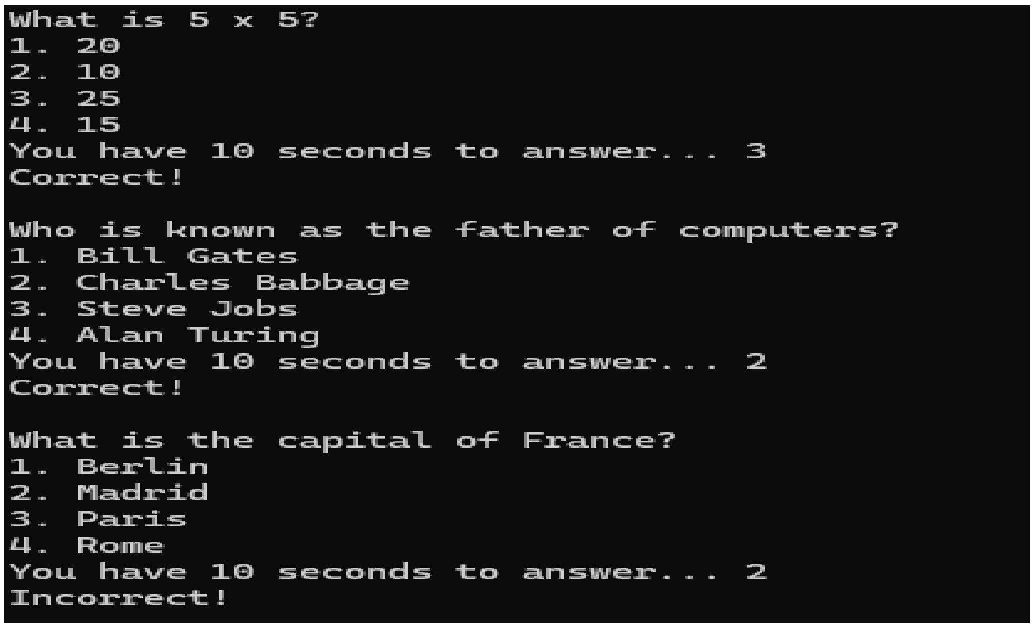
```
if (!answered) {
System.out.println("\nTime's up!");
summary.add("Q" + (i + 1) + ": Skipped");
} else if (q.isCorrect(userAnswer)) {
System.out.println("Correct!");
score++;
summary.add("Q" + (i + 1) + ": Correct");
} else {
System.out.println("Incorrect!");
summary.add("Q" + (i + 1) + ": Incorrect");
}}
System.out.println("\n=== Quiz Completed ===");
 System.out.println("Your Score: " + score + "/" + questions.size());
System.out.println("\nAnswer Summary:");
for (String result : summary) {
System.out.println(result);
}
sc.close();
}
}
```

Anuja Shiravale
Batch - APRIL BATCH B22

OUTPUT:-

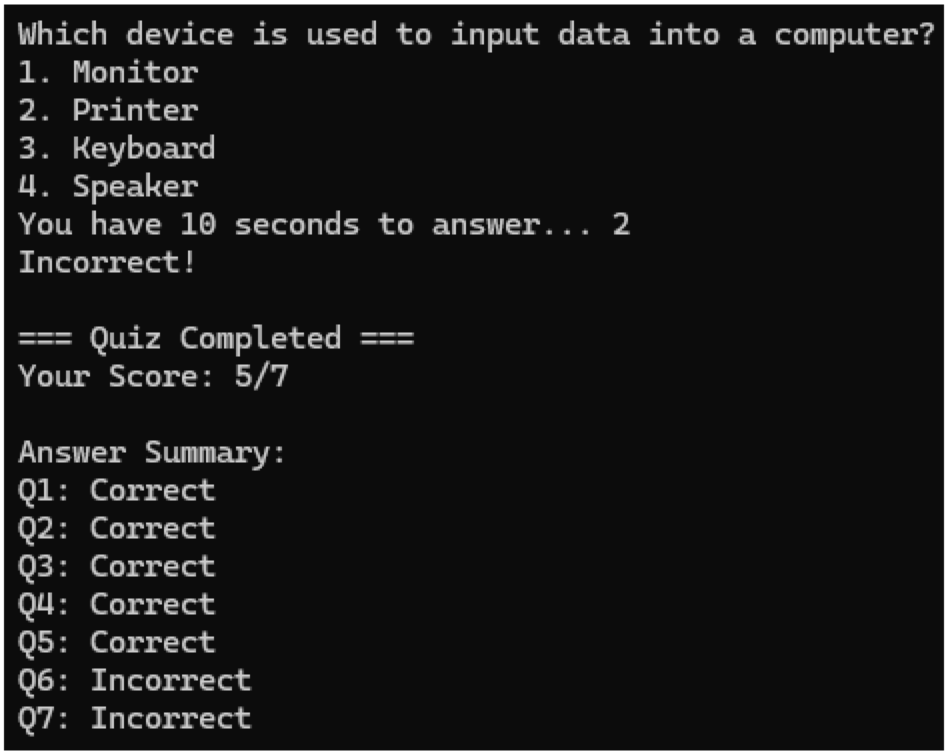


```
C:\1348>java QuizApp
Welcome to the Quiz! You have 10 seconds to answer each question.

Which language is used for Android app development?
1. Python
2. Java
3. Swift
4. C++
You have 10 seconds to answer... 2
Correct!

What does CPU stand for?
1. Central Program Unit
2. Central Processing Unit
3. Computer Power Unit
4. Control Processing Unit
You have 10 seconds to answer... 2
Correct!

Which planet is known as the Red Planet?
1. Earth
2. Venus
3. Mars
4. Jupiter
You have 10 seconds to answer... 3
Correct!
```



```
What is 5 x 5?
1. 20
2. 10
3. 25
4. 15
You have 10 seconds to answer... 3
Correct!

Who is known as the father of computers?
1. Bill Gates
2. Charles Babbage
3. Steve Jobs
4. Alan Turing
You have 10 seconds to answer... 2
Correct!

What is the capital of France?
1. Berlin
2. Madrid
3. Paris
4. Rome
You have 10 seconds to answer... 2
Incorrect!
```

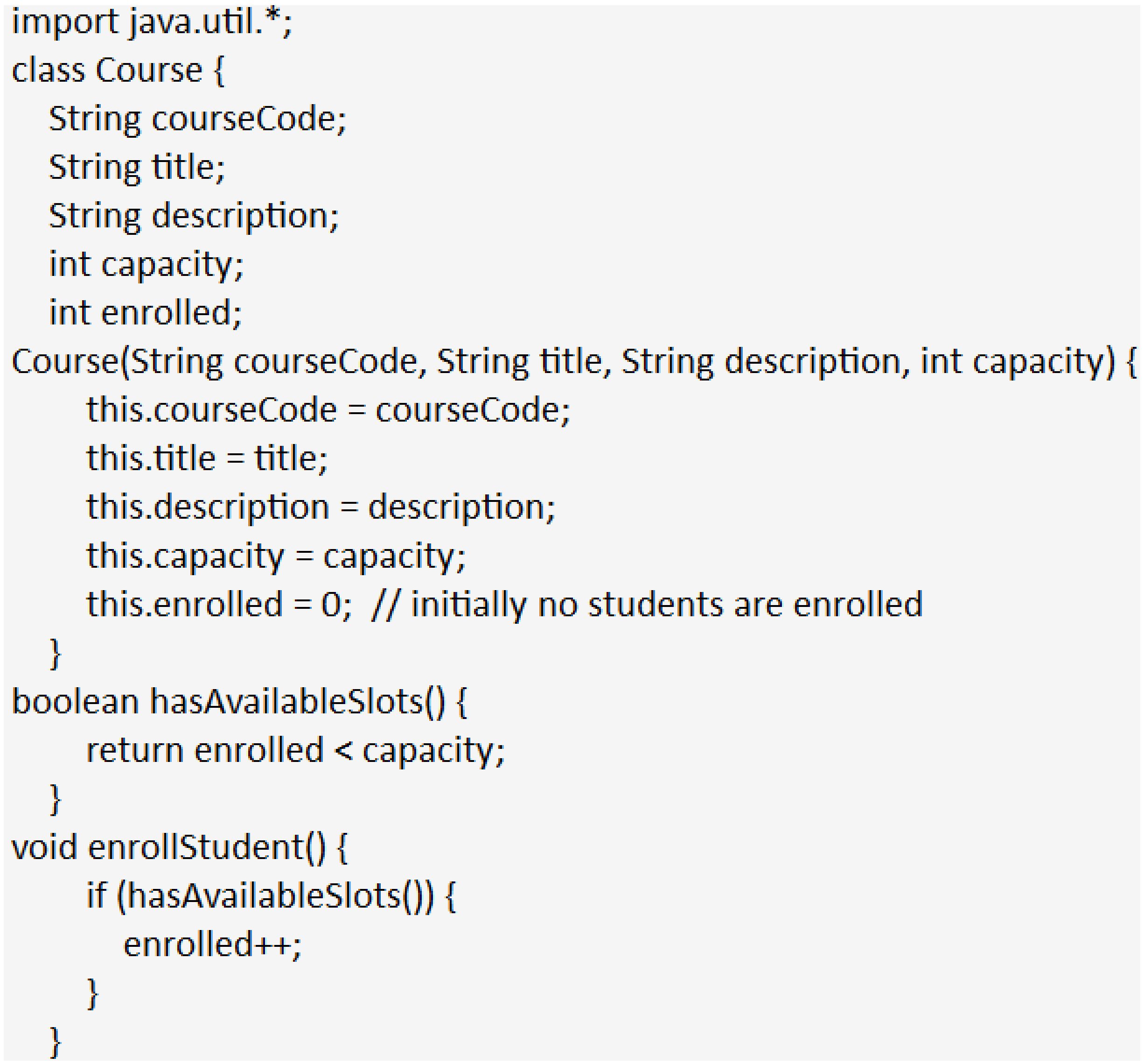
Anuja Shiravale
Batch - APRIL BATCH B22



```
Which device is used to input data into a computer?
1. Monitor
2. Printer
3. Keyboard
4. Speaker
You have 10 seconds to answer... 2
Incorrect!

=== Quiz Completed ===
Your Score: 5/7

Answer Summary:
Q1: Correct
Q2: Correct
Q3: Correct
Q4: Correct
Q5: Correct
Q6: Incorrect
Q7: Incorrect
```

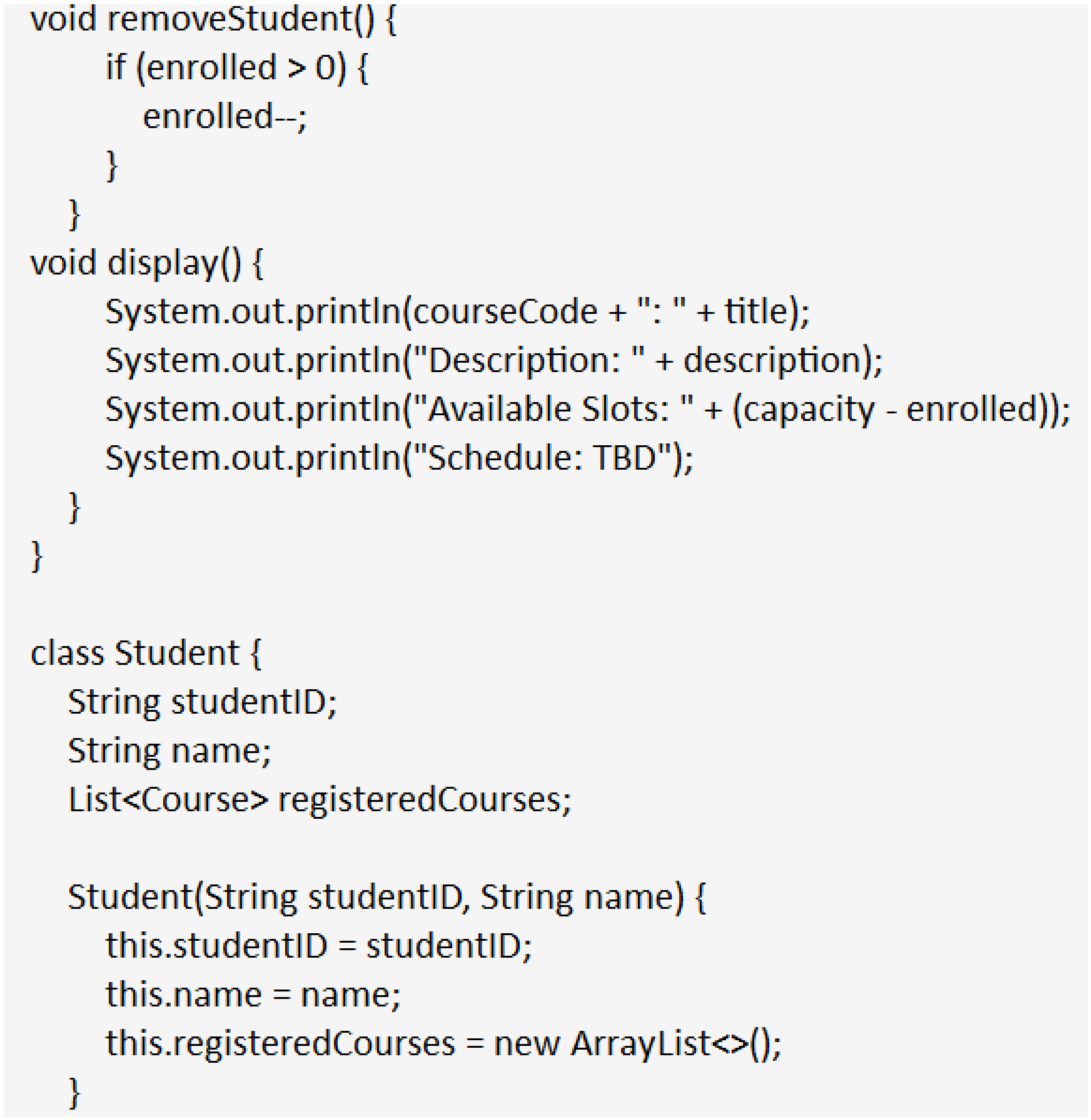
Anuja Shiravale
Batch - APRIL BATCH B22

TASK 5) STUDENT COURSE REGISTRATION SYSTEM



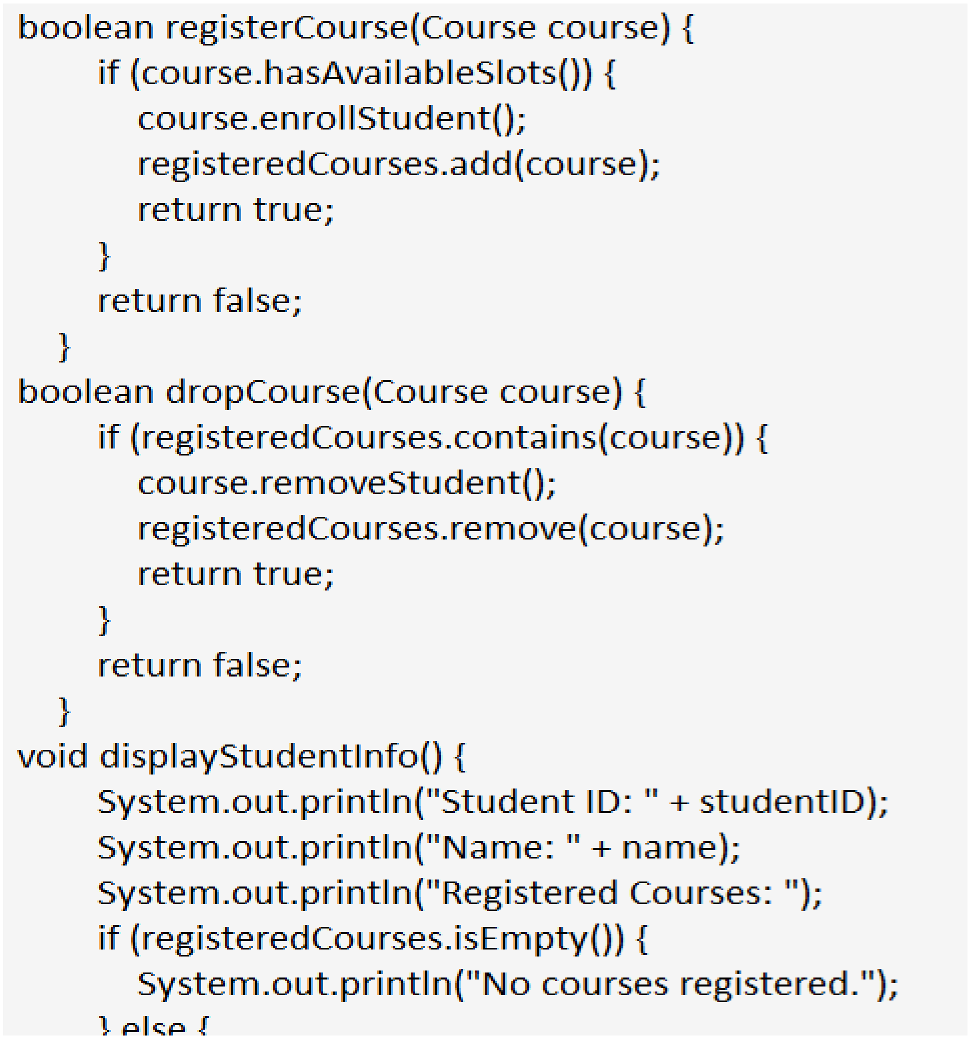
```
import java.util.*;
class Course {
    String courseCode;
    String title;
    String description;
    int capacity;
    int enrolled;
Course(String courseCode, String title, String description, int capacity) {
        this.courseCode = courseCode;
        this.title = title;
        this.description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

Anuja Shiravale
Batch - APRIL BATCH B22

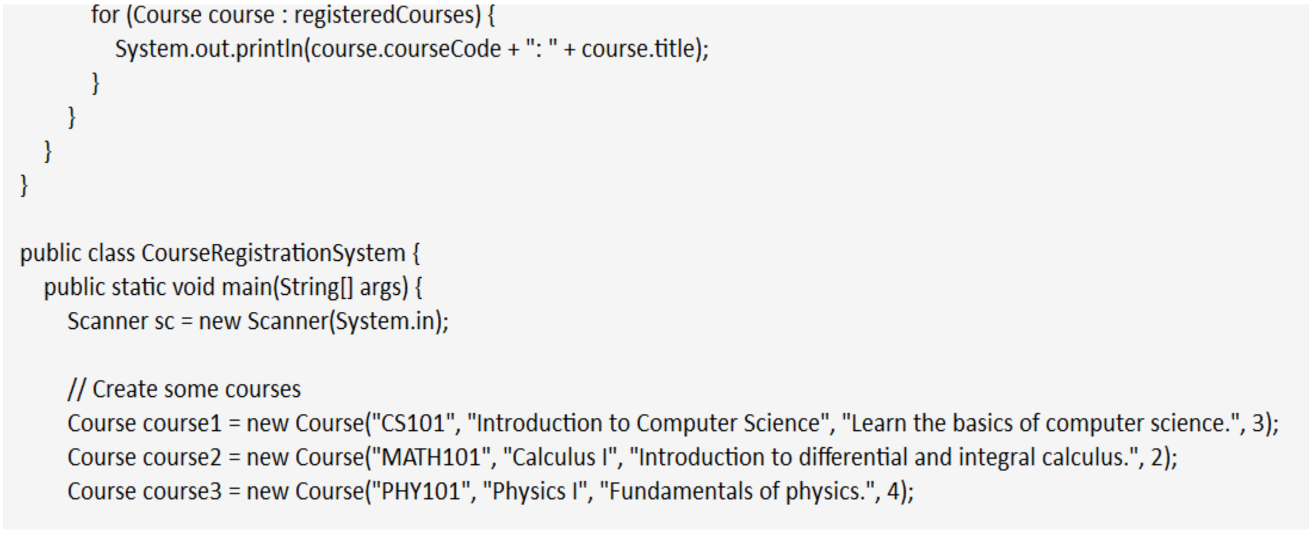


```
void removeStudent() {
        if (enrolled > 0) {
            enrolled--;
        }
    }
void display() {
        System.out.println(courseCode + ": " + title);
        System.out.println("Description: " + description);
        System.out.println("Available Slots: " + (capacity - enrolled));
 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

Anuja Shiravale
Batch - APRIL BATCH B22

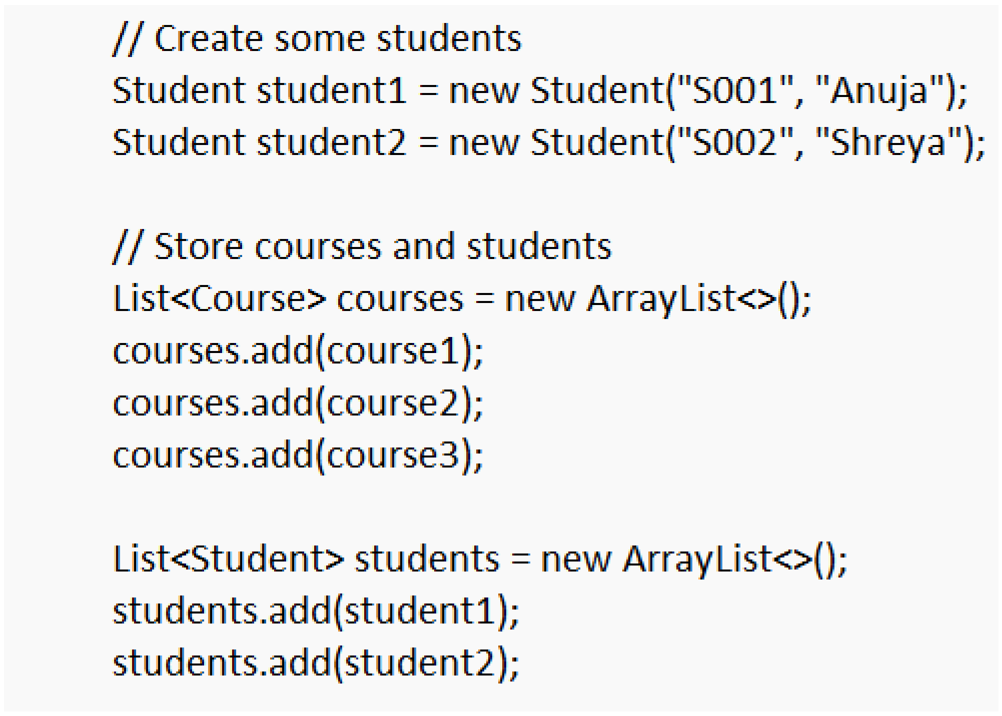


```
boolean registerCourse(Course course) {
        if (course.hasAvailableSlots()) {
            course.enrollStudent();
            registeredCourses.add(course);
            return true;
        }
        return false;
    }
boolean dropCourse(Course course) {
        if (registeredCourses.contains(course)) {
            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 {
```

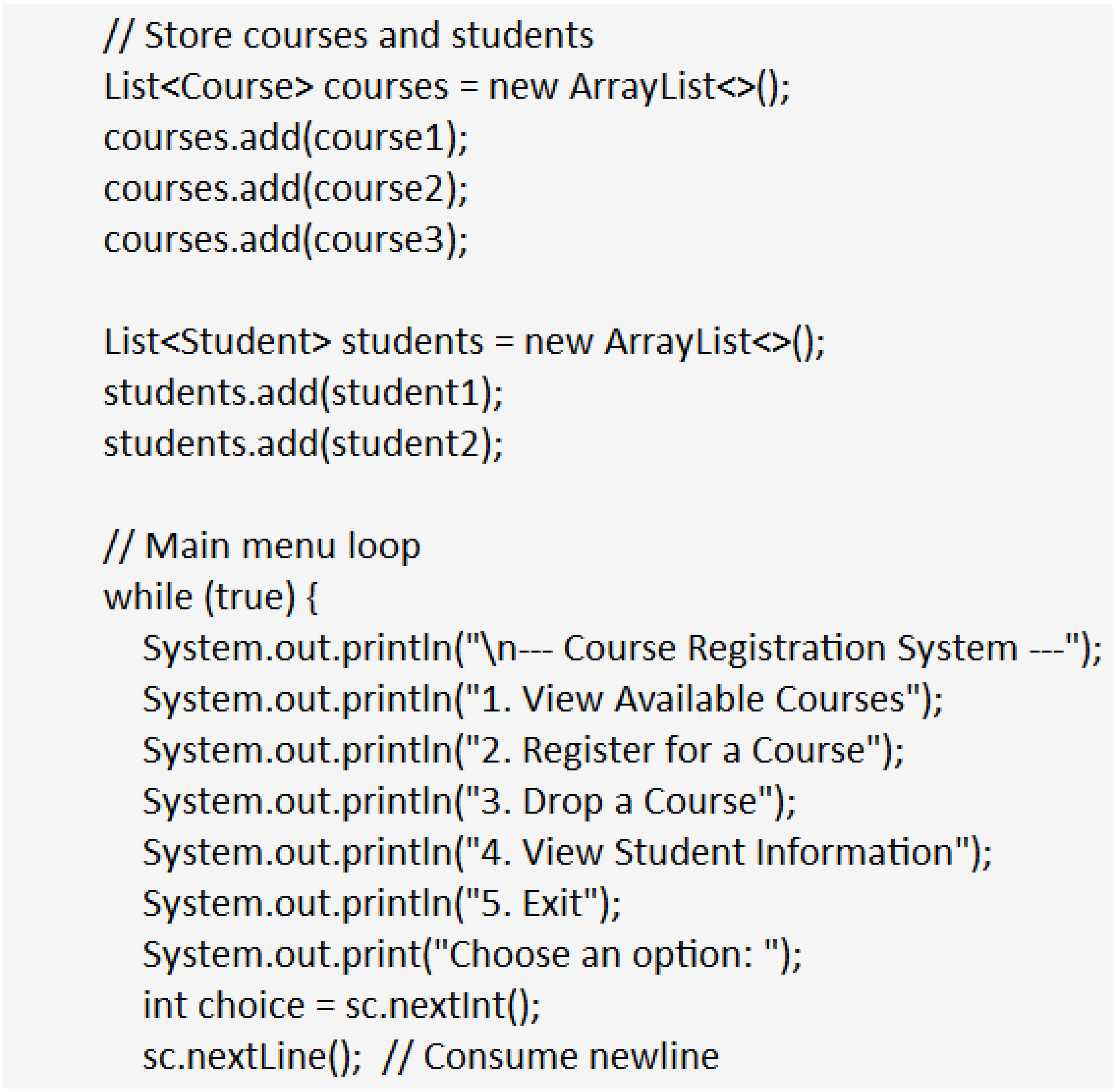


```
            for (Course course : registeredCourses) {
                System.out.println(course.courseCode + ": " + course.title);
            }
        }
    }
}

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);
```

Anuja Shiravale
Batch - APRIL BATCH B22



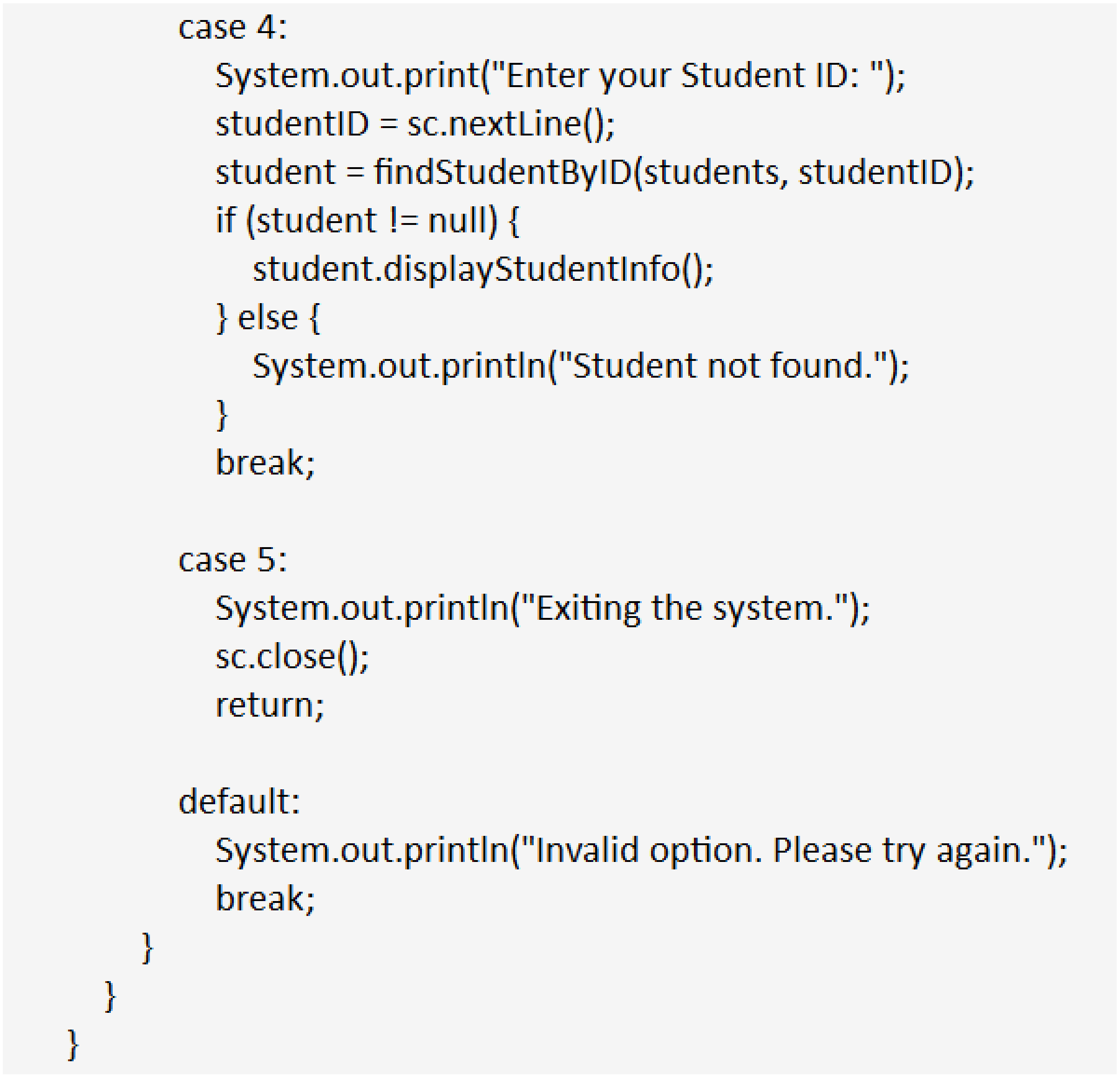
```
// Create some students
Student student1 = new Student("S001", "Anuja");
Student student2 = new Student("S002", "Shreya");

// 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;
```

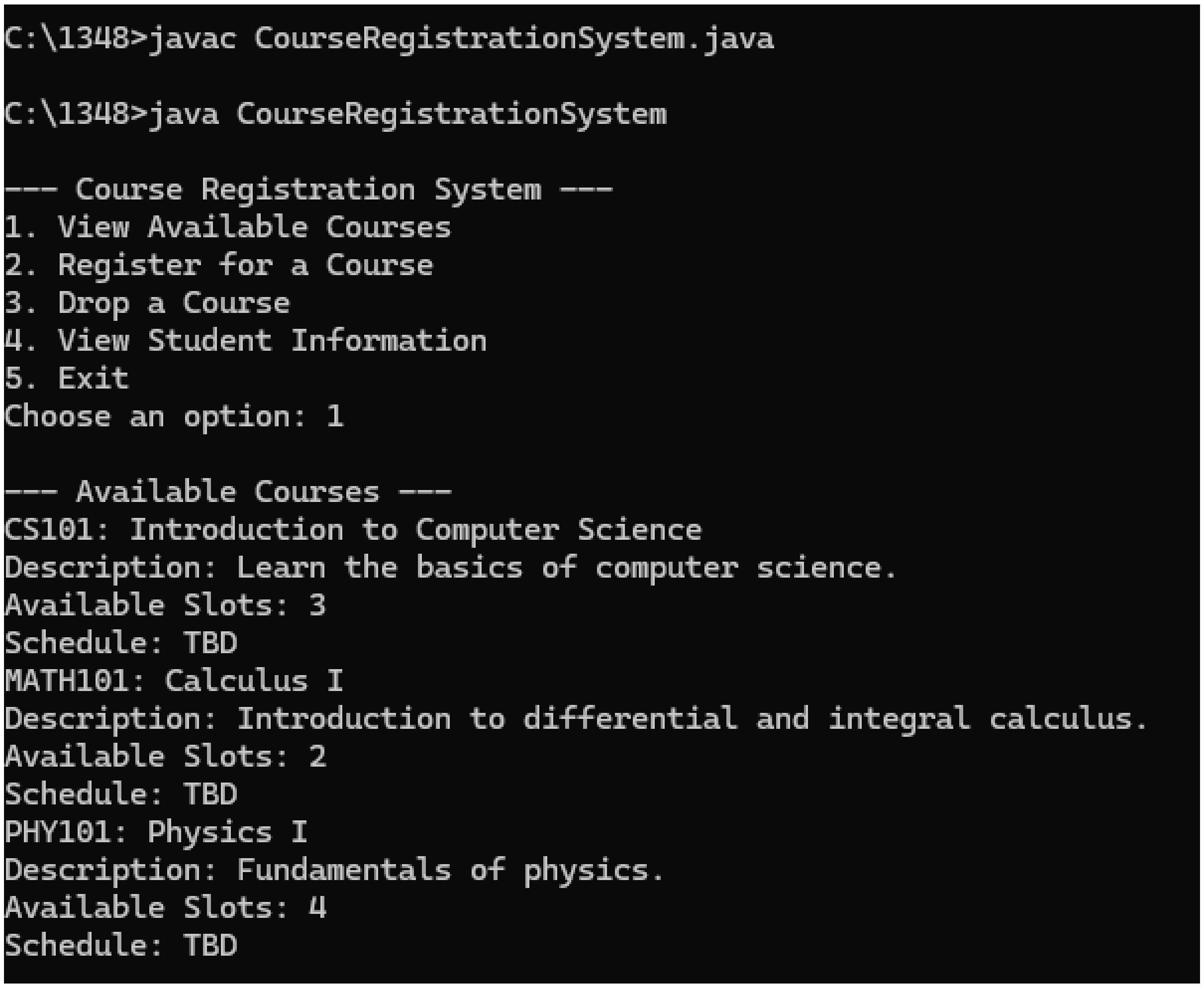


```
            case 4:
                System.out.print("Enter your Student ID: ");
                studentID = sc.nextLine();
                student = findStudentByID(students, studentID);
                if (student != null) {
                    student.displayStudentInfo();
                } else {
                    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;
    }
}
```

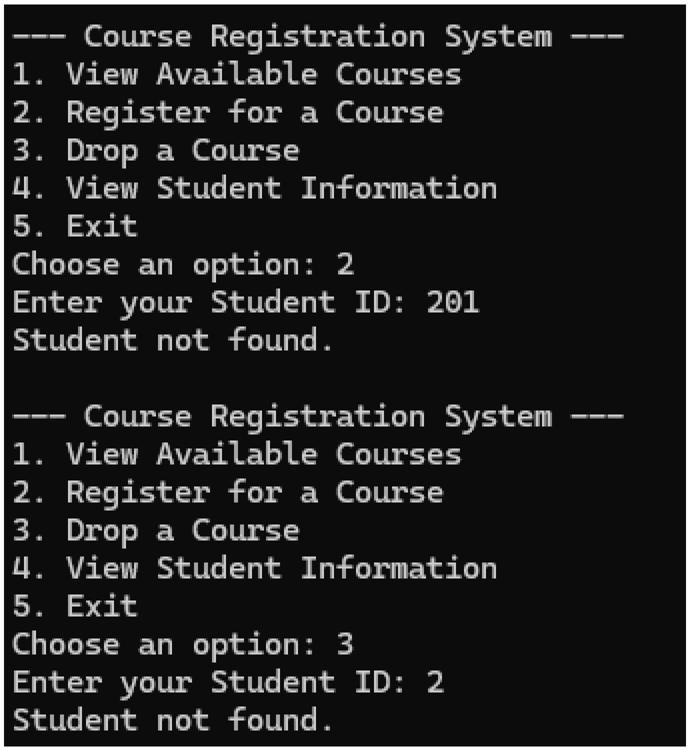
OUTPUT:-

Anuja Shiravale
Batch - APRIL BATCH B22

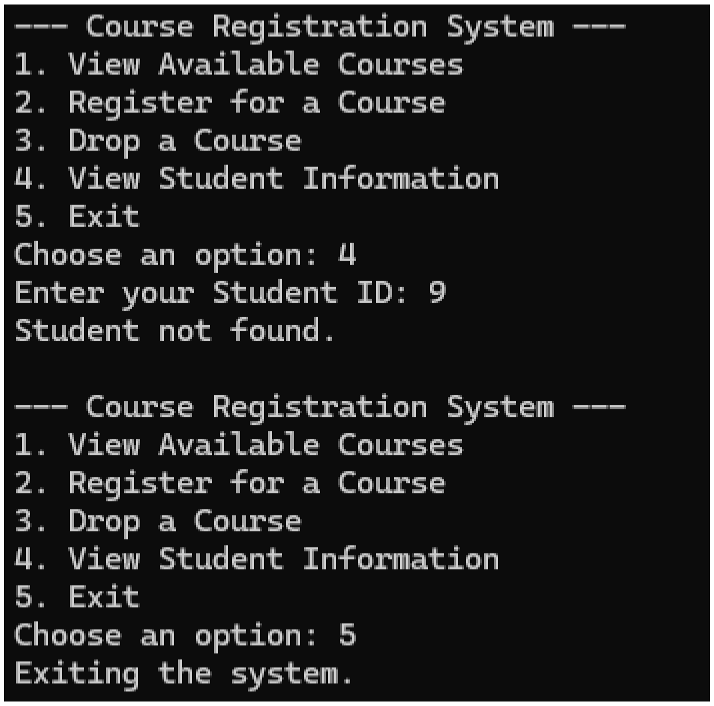


```
C:\1348>javac CourseRegistrationSystem.java

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
```

Anuja Shiravale
Batch - APRIL BATCH B22



```
--- 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.
```