TASK 5

STUDENT COURSE REGISTRATION SYSTEM

```
import java.util.*;
class Course {
   String code;
   String title;
   String description;
   int capacity;
   int slotsRemaining;
   String schedule;
   public Course(String code, String title, String description, int capacity, String schedule) {
      this.code = code;
      this.title = title;
      this.description = description;
      this.capacity = capacity;
      this.slotsRemaining = capacity;
      this.schedule = schedule;
   }
   @Override
   public String toString() {
      return String.format(
            "Course Code: %s\nTitle: %s\nDescription: %s\nCapacity: %d\nSlots Remaining: %d\nSchedule: %s\n",
            code, title, description, capacity, slotsRemaining, schedule);
   }
}
class Student {
   String id;
   String name;
```

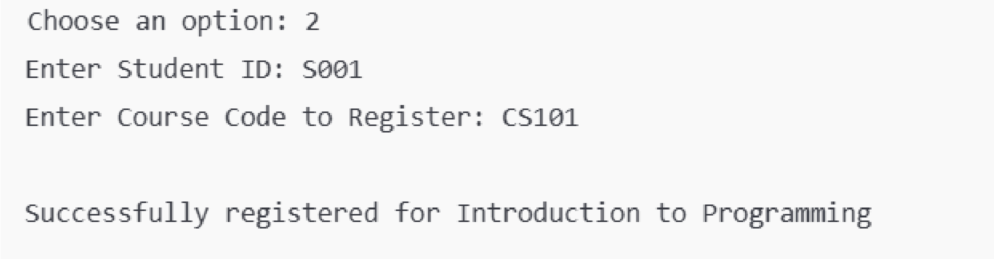
```
    List<String> registeredCourses;
    public Student(String id, String name) {
        this.id = id;
        this.name = name;
        this.registeredCourses = new ArrayList<>();
    }
    @Override
    public String toString() {
        return String.format("Student ID: %s\nName: %s\nRegistered Courses: %s\n", id, name,
registeredCourses);
    }
}
public class CourseRegistrationSystem {
   static Map<String, Course> courseDatabase = new HashMap<>();
    static Map<String, Student> studentDatabase = new HashMap<>();
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Sample data
        seedData();
        while (true) {
            System.out.println("\n*** Student Course Registration System ***");
            System.out.println("1. List Available Courses");
            System.out.println("2. Register for a Course");
            System.out.println("3. Drop a Course");
            System.out.println("4. View Student Details");
            System.out.println("5. Exit");
            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline
            switch (choice) {
```

```
        case 1:
          listCourses();
          break;
        case 2:
          registerForCourse(scanner);
          break;
        case 3:
          dropCourse(scanner);
          break;
        case 4:
          viewStudentDetails(scanner);
          break;
        case 5:
          System.out.println("Exiting... Thank you!");
          scanner.close();
          return;
        default:
          System.out.println("Invalid choice. Try again.");
      }
    }
  }
  static void seedData() {
    courseDatabase.put("CS101", new Course("CS101", "Introduction to Programming", "Learn basics of programming.", 30, "Mon-Wed 10AM-12PM"));
    courseDatabase.put("MATH101", new Course("MATH101", "Calculus I", "Introduction to differential calculus.", 25, "Tue-Thu 2PM-4PM"));
    courseDatabase.put("PHY101", new Course("PHY101", "Physics I", "Basic principles of physics.", 20, "Mon-Wed 1PM-3PM"));
    studentDatabase.put("S001", new Student("S001", "Alice Johnson"));
    studentDatabase.put("S002", new Student("S002", "Bob Smith"));
```

```
}
static void listCourses() {
    System.out.println("\n*** Available Courses ***");
    for (Course course : courseDatabase.values()) {
        System.out.println(course);
    }
}
static void registerForCourse(Scanner scanner) {
    System.out.print("Enter Student ID: ");
    String studentId = scanner.nextLine();
    Student student = studentDatabase.get(studentId);
    if (student == null) {
        System.out.println("Student not found!");
        return;
    }
    System.out.print("Enter Course Code to Register: ");
    String courseCode = scanner.nextLine();
    Course course = courseDatabase.get(courseCode);
    if (course == null) {
        System.out.println("Course not found!");
        return;
    }
    if (course.slotsRemaining <= 0) {
        System.out.println("No slots available for this course.");
        return;
    }
    if (student.registeredCourses.contains(courseCode)) {
        System.out.println("You are already registered for this course.");
        return;
        }
        student.registeredCourses.add(courseCode);
        course.slotsRemaining--;
        System.out.println("Successfully registered for " + course.title);
    }
    static void dropCourse(Scanner scanner) {
        System.out.print("Enter Student ID: ");
        String studentId = scanner.nextLine();
        Student student = studentDatabase.get(studentId);
        if (student == null) {
            System.out.println("Student not found!");
            return;
        }
        System.out.print("Enter Course Code to Drop: ");
        String courseCode = scanner.nextLine();
        Course course = courseDatabase.get(courseCode);
        if (course == null || !student.registeredCourses.contains(courseCode)) {
            System.out.println("You are not registered for this course.");
            return;
        }
        student.registeredCourses.remove(courseCode);
        course.slotsRemaining++;
        System.out.println("Successfully dropped " + course.title);
    }
    static void viewStudentDetails(Scanner scanner) {
        System.out.print("Enter Student ID: ");
        String studentId = scanner.nextLine();
        Student student = studentDatabase.get(studentId);
        if (student == null) {
```

```
      System.out.println("Student not found!");
      return;
    }
    System.out.println("\n*** Student Details ***");
    System.out.println(student);
  }
}
```

Output:



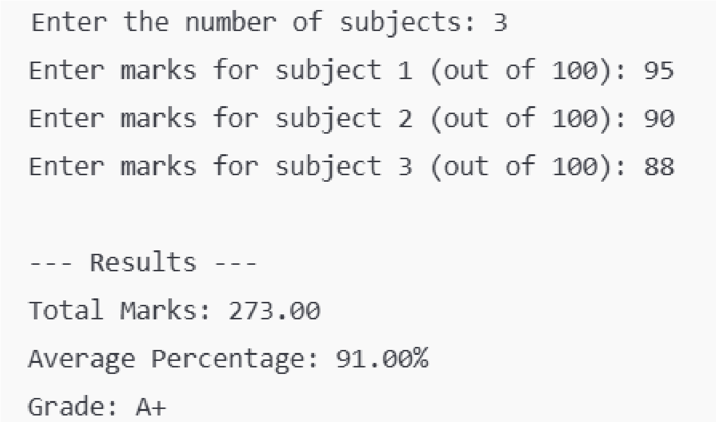
```
Choose an option: 2
Enter Student ID: S001
Enter Course Code to Register: CS101

Successfully registered for Introduction to Programming
```

TASK:2

STUDENT GRADE CALCULATOR

```
import java.util.Scanner;
public class StudentGradeCalculator {
  public static String calculateGrade(double percentage) {
    if (percentage >= 90) {
      return "A+";
    } else if (percentage >= 80) {
      return "A";
    } else if (percentage >= 70) {
      return "B";
    } else if (percentage >= 60) {
      return "C";
    } else if (percentage >= 50) {
      return "D";
    } else {
      return "F";
    }
  }

  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    // Input: Number of subjects
    System.out.print("Enter the number of subjects: ");
    int numSubjects = scanner.nextInt();
    // Input: Marks obtained in each subject
    double[] marks = new double[numSubjects];
    double totalMarks = 0;
    for (int i = 0; i < numSubjects; i++) {
            System.out.print("Enter marks for subject " + (i + 1) + " (out of 100): ");
            marks[i] = scanner.nextDouble();
            totalMarks += marks[i];
        }
        double averagePercentage = totalMarks / numSubjects;
        String grade = calculateGrade(averagePercentage);
        System.out.println("\n--- Results ---");
        System.out.printf("Total Marks: %.2f\n", totalMarks);
        System.out.printf("Average Percentage: %.2f%%\n", averagePercentage);
        System.out.println("Grade: " + grade);
        scanner.close();
    }
}
```

Output:



```
Enter the number of subjects: 3
Enter marks for subject 1 (out of 100): 95
Enter marks for subject 2 (out of 100): 90
Enter marks for subject 3 (out of 100): 88

--- Results ---
Total Marks: 273.00
Average Percentage: 91.00%
Grade: A+
```

TASK:4

QUIZ APPLICATION WITH TIMER

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:id="@+id/idLLSCORE"
  android:orientation="vertical">

  <TextView
    android:id="@+id/idTVSCORE"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:gravity="center"
    android:padding="3dp"
    android:text="YOUR SCORE IS"
    android:textAlignment="center"
    android:textAllCaps="false"
    android:textColor="@color/black"
    android:textSize="20dp"
    android:textStyle="bold" />

  <Button
    android:id="@+id/idBtnRestart"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="20dp"
    android:text="Restart Quiz"
    android:textAllCaps="false" />
</LinearLayout>
```

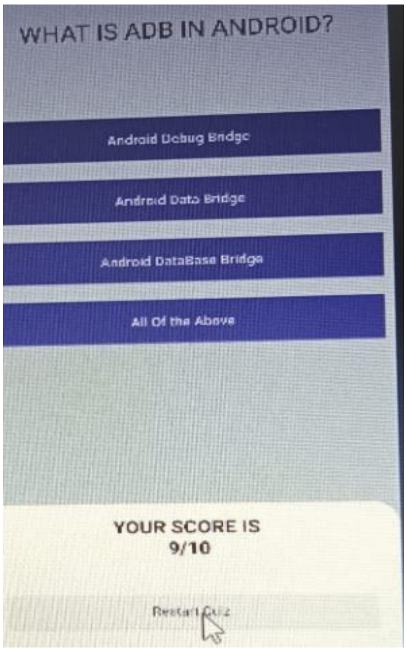
Main activity

```
import android.annotation.SuppressLint;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import com.google.android.material.bottomsheet.BottomSheetDialog;
import java.util.ArrayList;
import java.util.Random;
public class MainActivity extends AppCompatActivity {
  private TextView questionTV,questionNumberTV;
  private Button option1Btn,option2Btn,option3Btn,option4Btn;
  private ArrayList<QuizModel> quizModelArrayList;
  Random random;
  int currentScore = 0, questionAttempted = 1, currentPos;
  private ViewGroup LinearLayout;
  @SuppressLint("MissingInflatedId")
  @Override
  protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        questionTV = findViewById(R.id.idTVQuestion);
        questionNumberTV = findViewById(R.id.idTVQuestionAttempted);
        option1Btn = findViewById(R.id.idBtnOption1);
        option2Btn = findViewById(R.id.idBtnOption2);
        option3Btn = findViewById(R.id.idBtnOption3);
        option4Btn = findViewById(R.id.idBtnOption4);
        quizModelArrayList = new ArrayList<>();
        random = new Random();
        getQuizQuestion(quizModelArrayList);
        currentPos = random.nextInt(quizModelArrayList.size());
        setDataToViews(currentPos);
        option1Btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (quizModelArrayList.get(currentPos).getAnswer().trim().toLowerCase().equals(option1Btn.getText().toString().trim().toLowerCase())){
                    currentScore++;
                }
                questionAttempted++;
                currentPos = random.nextInt(quizModelArrayList.size());
                setDataToViews(currentPos);
            }
        });
        option2Btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (quizModelArrayList.get(currentPos).getAnswer().trim().toLowerCase().equals(option2Btn.getText().toString().trim().toLowerCase())){
                    currentScore++;
                }
                questionAttempted++;
                currentPos = random.nextInt(quizModelArrayList.size());
                setDataToViews(currentPos);

            }
        });
        option3Btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (quizModelArrayList.get(currentPos).getAnswer().trim().toLowerCase().equals(option3Btn.getText().toString().trim().toLowerCase())){
                    currentScore++;
                }
                questionAttempted++;
                currentPos = random.nextInt(quizModelArrayList.size());
                setDataToViews(currentPos);

            }
        });
        option4Btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (quizModelArrayList.get(currentPos).getAnswer().trim().toLowerCase().equals(option4Btn.getText().toString().trim().toLowerCase())){
```

```
                currentScore++;
            }
            questionAttempted++;
            currentPos = random.nextInt(quizModelArrayList.size());
            setDataToViews(currentPos);
        }
    });
}
private void showBottomSheet(){
    BottomSheetDialog bottomSheetDialog=new BottomSheetDialog(MainActivity.this);
    View bottomSheetView =
LayoutInflater.from(getApplicationContext()).inflate(R.layout.score_bottom_sheet,(LinearLay
out)findViewById(R.id.idLLSCORE));
    TextView SCORETV=bottomSheetView.findViewById(R.id.idTVSCORE);
    Button restartQuizBtn=bottomSheetView.findViewById(R.id.idBtnRestart);
    SCORETV.setText("YOUR SCORE IS \n"+currentScore+"/10");
    restartQuizBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            currentPos= random.nextInt(quizModelArrayList.size());
            setDataToViews(currentPos);
            questionAttempted=1;
            currentScore=0;
            bottomSheetDialog.dismiss();
        }
    });
    bottomSheetDialog.setCancelable(false);
    bottomSheetDialog.setContentView(bottomSheetView);
    bottomSheetDialog.show();
}
```

```
    private void setDataToViews(int currentPos) {
        questionNumberTV.setText("Questions Attempted : "+questionAttempted + "/10");
        if(questionAttempted==10){
            showBottomSheet();
        }
        else {
            questionTV.setText(quizModelArrayList.get(currentPos).getQuestion());
            option1Btn.setText(quizModelArrayList.get(currentPos).getOption1());
            option2Btn.setText(quizModelArrayList.get(currentPos).getOption2());
            option3Btn.setText(quizModelArrayList.get(currentPos).getOption3());
            option4Btn.setText(quizModelArrayList.get(currentPos).getOption4());
        }
    }
    private void getQuizQuestion(ArrayList<QuizModel> quizModelArrayList) {
        quizModelArrayList.add(new QuizModel("HOW GFD IS USED?", "To solve DSA problems","To learn New language","To learn Android","All Of the Above","All Of the Above"));
        quizModelArrayList.add(new QuizModel("WHAT IS GCM IN ANDROID?", "Google Cloud Messaging","Google Message Pack","Google Cloud Manager","All Of the Above","Google Cloud Messaging"));
        quizModelArrayList.add(new QuizModel("WHAT IS ADB IN ANDROID?", "Android Debug Bridge","Android Data Bridge","Android DataBase Bridge","All Of the Above","Android Debug Bridge"));
        quizModelArrayList.add(new QuizModel("WHAT ARE THE COLOUR PRESENT IN ANDROID?", "Colors.xml","AndroidManifest.xml","Strings.xml","All Of the Above","Colors.xml"));
```

OUTPUT:



WHAT IS ADB IN ANDROID?

YOUR SCORE IS
9/10