

Programming in Java MAT272

*Department of Computer Science
Christ(deemed-to-be) University
Bengaluru-560029*

NAME: Pranab Rai
REG NO: 2447137

Lab-exercise:1

CODE:

```
public class Course {
    private String courseName;
    private double courseDuration;
    private String tutorName;
    private String studentName;
    private int studentAge;

    // Static variable t
    private static int totalStudents = 0;

    // Constructor
    public Course(String courseName, String tutorName) {
        this.courseName = courseName;
        this.tutorName = tutorName;
    }

    // Overloaded constructor
    public Course(String courseName, double courseDuration, String tutorName, String studentName, int studentAge) {
        this.courseName = courseName;
        this.courseDuration = courseDuration;
        this.tutorName = tutorName;
        this.studentName = studentName;
        this.studentAge = studentAge;
        totalStudents++;
    }

    // Static method
    public static int getTotalStudents() {
        return totalStudents;
    }

    // Method to display course and student details
    public void displayDetails() {
        System.out.println("Course Name: " + courseName);
        if (courseDuration > 0) {
```

```

        System.out.println("Course Duration: " + courseDuration + " hours");
    }
    System.out.println("Tutor Name: " + tutorName);
    if (studentName != null) {
        System.out.println("Student Name: " + studentName);
        System.out.println("Student Age: " + studentAge);
    }
}

// Overloaded static method- by course name
public static void displayDetails(Course[] courses, String courseName) {
    for (Course course : courses) {
        if (course.courseName.equals(courseName)) {
            course.displayDetails();
            System.out.println("-----");
        }
    }
}

// Static method- by tutor name
public static void displayDetailsByTutor(Course[] courses, String tutorName) {
    for (Course course : courses) {
        if (course.tutorName.equals(tutorName)) {
            course.displayDetails();
            System.out.println("-----");
        }
    }
}

// Static method to reset no of students
public static void resetTotalStudents() {
    totalStudents = 0;
}

// StringBuilder
public static String getCourseDurations(Course[] courses) {
    StringBuilder sb = new StringBuilder();
    for (Course course : courses) {
        //method chaining
        sb.append(course.courseDuration).append(" hours, ");
    }
    //subset of sb
    return sb.substring(0, sb.length() - 2);
}

public static void main(String[] args) {
    Course course1 = new Course("Web Stack Development", 75, "Cynthia T", "Lara", 20);
    Course course2 = new Course("Software Engineering", 45, "Neha Singhal", "Bobby", 22);
}

```

```

    Course course3 = new Course("Problem Solving using C", 90, "Dr. Shoney Sebastian", "Timmy",
23);

    Course[] courses = { course1, course2, course3 };

    // Display all course details
    System.out.println("Displaying all courses:");
    for (Course course : courses) {
        course.displayDetails();
        System.out.println("-----");
    }

    // Display specific courses
    System.out.println("Displaying courses with name 'Software Engineering':");
    Course.displayDetails(courses, "Software Engineering");

    System.out.println("Displaying courses by tutor 'Cynthia T:");
    Course.displayDetailsByTutor(courses, "Cynthia T");

    System.out.println("Total Students Enrolled: " + Course.getTotalStudents());

    // Display StringBuilder
    System.out.println("Course Names: " + course1.courseName + ", " + course2.courseName + ", " +
course3.courseName);
    System.out.println("Course total duration: " + Course.getCourseDurations(courses));
}
}

```

SUMMARY

1. **Course Class:** It holds details like course name, duration, tutor name, and student info (name, age). The class also tracks the total number of students enrolled using a static variable.
2. **Constructors:** The overloaded constructor is used to create course objects with course and student details.
3. **Static Methods:** A static method `getTotalStudents()` tracks and returns the total number of students. `displayDetails()` is overloaded to allow displaying courses by course name or by tutor name.
4. **Main Method:** In the main method, three course objects are created with full details. It demonstrates how to display all course details, search by course name, and search by tutor name. Additionally, a static variable is used to print the total number of students.
5. **Output:** The program prints details of all the courses, filters courses by name and tutor, and shows the total number of students enrolled. It uses the String Builder types to output the results.

OUTPUT

```
Displaying all courses:
Course Name: Web Stack Development
Course Duration: 75.0 hours
Tutor Name: Cynthia T
Student Name: Lara
Student Age: 20
-----
Course Name: Software Engineering
Course Duration: 45.0 hours
Tutor Name: Neha Singhal
Student Name: Bobby
Student Age: 22
-----
Course Name: Problem Solving using C
Course Duration: 90.0 hours
Tutor Name: Dr. Shoney Sebastian
Student Name: Timmy
Student Age: 23
-----
Displaying courses with name 'Software Engineering':
Course Name: Software Engineering
Course Duration: 45.0 hours
Tutor Name: Neha Singhal
Student Name: Bobby
Student Age: 22
-----
Displaying courses by tutor 'Cynthia T':
Course Name: Web Stack Development
Course Duration: 75.0 hours
Tutor Name: Cynthia T
Student Name: Lara
Student Age: 20
-----
Total Students Enrolled: 3
Course Names: Web Stack Development, Software Engineering, Problem Solving using C
Course total duration: 75.0 hours, 45.0 hours, 90.0 hours
```