

# **K.R. Mangalam University**

School of Engineering & Technology

## **Lab Assignment 1**

Java Programming

Submitted by:

Name: Rakesh G

Roll No: 2401201064

Class: BCA (AI & DS)

Submitted to:

Dr. Manish Kumar

GitHub Repository:

[https://github.com/rakesh4407/Java\\_Assignment\\_Rakesh](https://github.com/rakesh4407/Java_Assignment_Rakesh)

# CODE:

D:\RAKESH\VS\JAVA\practice.java recordSystem.java > Language Support for Java(TM) by Red Hat > Student > Student()

```
1  import java.util.ArrayList;
2  import java.util.Scanner;
3
4  // Base Class
5  class Person {
6
7      String name;
8
9      Person() {
10         this.name = "";
11     }
12
13     Person(String name) {
14         this.name = name;
15     }
16 }
17
18 // Derived Class: Student
19 class Student extends Person {
20
21     int rollNo;
22     String course;
23     double marks;
24     char grade;
25
26     // Default Constructor
27     Student() {
28         super();
29         this.rollNo = 0;
30         this.course = "";
31         this.marks = 0.0;
32         this.grade = 'F';
33     }
34
35     // Parameterized Constructor
36     Student(int rollNo, String name, String course, double marks) { Constructor is never used
37         super(name);
38         this.rollNo = rollNo;
```

JAVA > Assignment > StudentRecordSystem.java > Language Support for Java(TM) by Red Hat > Student > Student()

```
19  class Student extends Person {
36      Student(int rollNo, String name, String course, double marks) { Constructor is never used
40          this.marks = marks;
41          calculateGrade(); Overridable method call in constructor
42      }
43
44      // Method to take input from user
45      void inputDetails(Scanner sc) {
46          System.out.print(s:"Enter Roll No: ");
47          rollNo = sc.nextInt();
48          sc.nextLine(); // consume newline
49
50          System.out.print(s:"Enter Name: ");
51          name = sc.nextLine();
52
53          System.out.print(s:"Enter Course: ");
54          course = sc.nextLine();
55
56          System.out.print(s:"Enter Marks (0-100): ");
57          marks = sc.nextDouble();
58
59          // Validate marks
60          while (marks < 0 || marks > 100) {
61              System.out.print(s:"Invalid Marks! Enter again (0-100): ");
62              marks = sc.nextDouble();
63          }
64
65          calculateGrade();
66      }
67
68      // Method to calculate grade
69      void calculateGrade() {
70          if (marks >= 85) {
71              grade = 'A';
72          } else if (marks >= 70) {
73              grade = 'B';
74          } else if (marks >= 50) {
75              grade = 'C';
```

JAVA > Assingment > StudentRecordSystem.java > Language Support for Java(TM) by Red Hat > Student > Student()

```
19 class Student extends Person {
69     void calculateGrade() {
77         grade = 'D';
78     }
79 }
80
81 // Method to display details
82 void displayDetails() {
83     System.out.println("Roll No: " + rollNo);
84     System.out.println("Name: " + name);
85     System.out.println("Course: " + course);
86     System.out.println("Marks: " + marks);
87     System.out.println("Grade: " + grade);
88     System.out.println(x:"-----");
89 }
90 }
91
92 // Main Class with Menu
93 public class StudentRecordSystem {
94
95     Run main | Debug main | Run | Debug
96     public static void main(String[] args) {
97         Scanner sc = new Scanner(System.in);    Convert to try-with-resources
98         ArrayList<Student> students = new ArrayList<>();
99         int choice;
100
101         do {
102             System.out.println(x:"==== Student Record Menu =====");
103             System.out.println(x:"1. Add Student");
104             System.out.println(x:"2. Display All Students");
105             System.out.println(x:"3. Exit");
106             System.out.print(s:"Enter your choice: ");
107             choice = sc.nextInt();
108
109             switch (choice) {    Convert switch to rule switch
110                 case 1:
111                     Student s = new Student();
112                     s.inputDetails(sc);
113 
```

JAVA > Assingment > StudentRecordSystem.java > Language Support for Java(TM) by Red Hat > Student > Student()

```
93 public class StudentRecordSystem {
95     public static void main(String[] args) {
105         System.out.print(s:"Enter your choice: ");
106         choice = sc.nextInt();
107
108         switch (choice) {    Convert switch to rule switch
109             case 1:
110                 Student s = new Student();
111                 s.inputDetails(sc);
112                 students.add(s);
113                 System.out.println(x:"Student Added Successfully!\n");
114                 break;
115
116             case 2:
117                 if (students.isEmpty()) {
118                     System.out.println(x:"No student records available.\n");
119                 } else {
120                     System.out.println(x:"==== Student Records =====");
121                     for (Student st : students) {
122                         st.displayDetails();
123                     }
124                 }
125                 break;
126
127             case 3:
128                 System.out.println(x:"Exiting the application. Goodbye!");
129                 break;
130
131             default:
132                 System.out.println(x:"Invalid choice. Please try again.\n");
133         }
134     } while (choice != 3);
135
136     sc.close();
137 }
138 }
139 
```

# OUTPUT:

```
===== Student Record Menu =====
1. Add Student
2. Display All Students
3. Exit
Enter your choice: 1
Enter Roll No: 64
Enter Name: Rakesh
Enter Course: BCA
Enter Marks (0-100): 99
Student Added Successfully!

===== Student Record Menu =====
1. Add Student
2. Display All Students
3. Exit
Enter your choice: 2
===== Student Records =====
Roll No: 64
Name: Rakesh
Course: BCA
Marks: 99.0
Grade: A
-----
===== Student Record Menu =====
1. Add Student
2. Display All Students
3. Exit
Enter your choice: 3
Exiting the application. Goodbye!

D:\RAKESH\VSC>
```