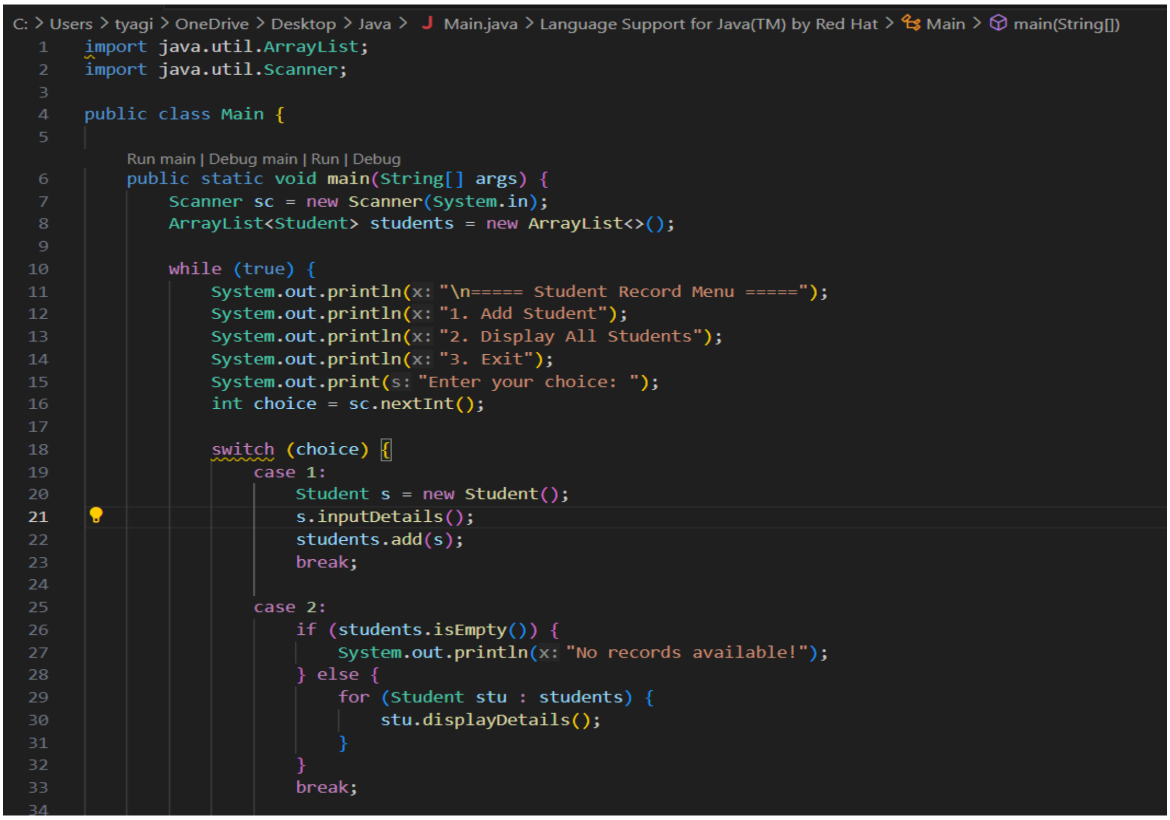
Java Lab Assignment 1

NAME – SHASHANK TYAGI

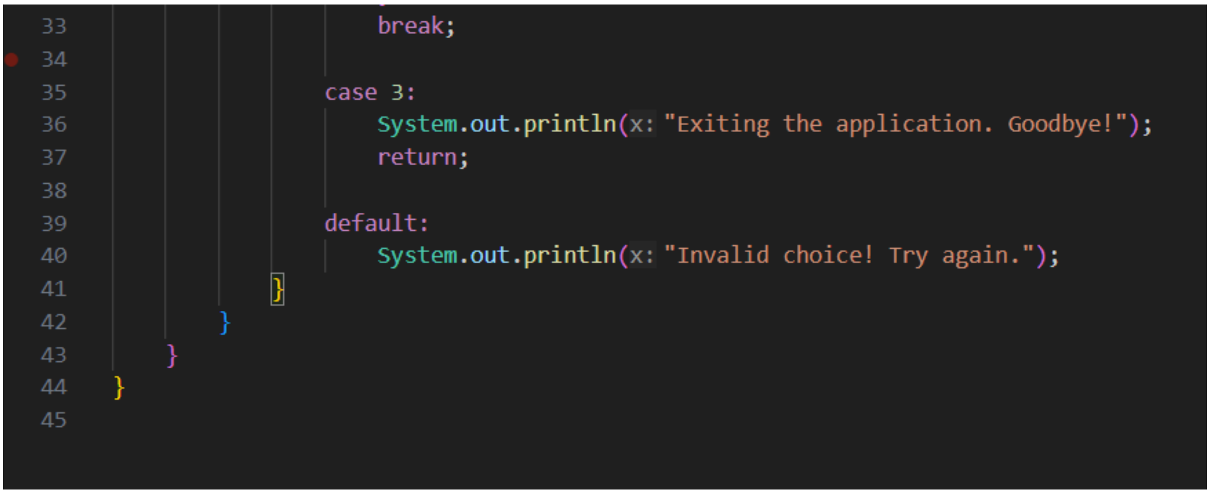
ROLL NO – 2401201033

COURSE- BCA(AI&DS)

code -

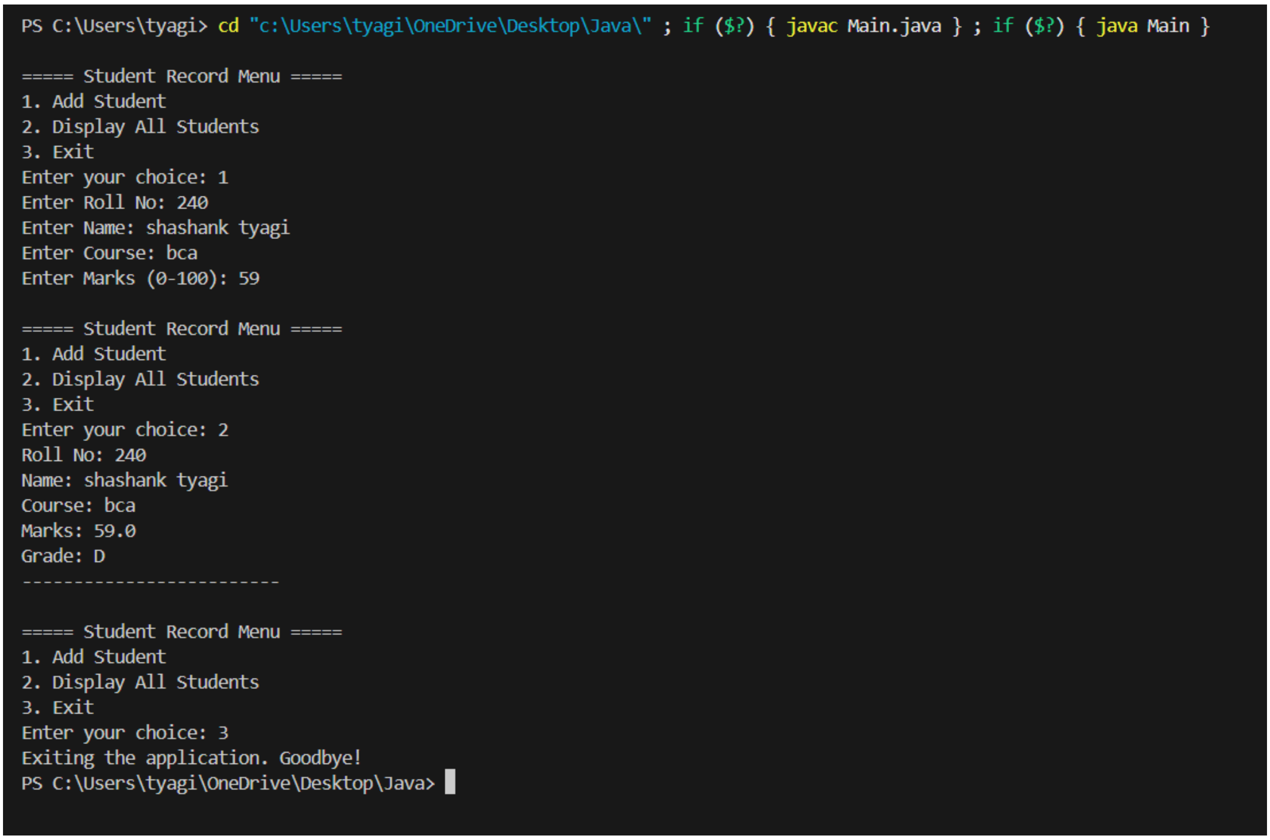

C: > Users > tyagi > OneDrive > Desktop > Java > Main.java > Language Support for Java(TM) by Red Hat > Main > main(String[])
Run main | Debug main | Run | Debug

```java
import java.util.ArrayList;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        ArrayList<Student> students = new ArrayList<>();

        while (true) {
            System.out.println(x: "\n===== Student Record Menu =====");
            System.out.println(x: "1. Add Student");
            System.out.println(x: "2. Display All Students");
            System.out.println(x: "3. Exit");
            System.out.print(s: "Enter your choice: ");
            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    Student s = new Student();
                    s.inputDetails();
                    students.add(s);
                    break;

                case 2:
                    if (students.isEmpty()) {
                        System.out.println(x: "No records available!");
                    } else {
                        for (Student stu : students) {
                            stu.displayDetails();
                        }
                    }
                    break;

```



```java
                    break;

                case 3:
                    System.out.println(x: "Exiting the application. Goodbye!");
                    return;

                default:
                    System.out.println(x: "Invalid choice! Try again.");
            }
        }
    }
}

```

Output -



```
PS C:\Users\tyagi> cd "c:\Users\tyagi\OneDrive\Desktop\Java\" ; if ($?) { javac Main.java } ; if ($?) { java Main }

===== Student Record Menu =====
1. Add Student
2. Display All Students
3. Exit
Enter your choice: 1
Enter Roll No: 240
Enter Name: shashank tyagi
Enter Course: bca
Enter Marks (0-100): 59

===== Student Record Menu =====
1. Add Student
2. Display All Students
3. Exit
Enter your choice: 2
Roll No: 240
Name: shashank tyagi
Course: bca
Marks: 59.0
Grade: D
-------------------------

===== Student Record Menu =====
1. Add Student
2. Display All Students
3. Exit
Enter your choice: 3
Exiting the application. Goodbye!
PS C:\Users\tyagi\OneDrive\Desktop\Java>
```

EXPLANATION –

The Main class works as the central controller of the Student Record Management System. It handles user interaction, menu display, and communication with the Student objects. The program begins by creating a Scanner object to read input from the keyboard and an ArrayList to store multiple student records. The use of an ArrayList allows dynamic storage because students can be added without specifying any fixed size.

The program enters an infinite loop that continuously displays a menu to the user. The menu provides three options:

1. **Add a new student,**
2. **Display all stored student records, and**
3. **Exit the application.**

This loop ensures that the program keeps running until the user chooses to exit.

When the user selects an option, the program reads the choice and uses a switch statement to handle the selected operation. If the user chooses the option to add a student, the program creates a new Student object and calls a method that prompts the user to input all required details such as roll number, name, course, and marks. After collecting the information, the student object is stored in the ArrayList.

If the user chooses to display all students, the program first checks whether the list is empty. If no records exist, an appropriate message is shown. Otherwise, the program loops through the ArrayList and displays the details of each student. This is done by calling a method from the Student class that prints the stored information along with the calculated grade.

When the user selects the exit option, the program prints a closing message and terminates the loop, effectively stopping the application. If the user enters any invalid input (anything other than 1, 2, or 3), a message is shown informing them that the choice is invalid, and the menu is displayed again.

Overall, the Main program manages user interaction, directs program flow using looping and decision-making structures, and coordinates actions between the user and the Student objects. It demonstrates the practical use of object-oriented programming, collections, and control structures in Java.