

# CSA0961 – JAVA

## PRACTISE – 5

### JF section 5 Practice

```
import java.util.Scanner;

public class ColorRange {

    public static void main(String[] args) {

        // Create a Scanner object for user input

        Scanner scanner = new Scanner(System.in);

        // Define the valid range for each color component

        int minRange = 0;

        int maxRange = 255;

        // Prompt user to enter RGB values

        System.out.print("<div data-bbox='\">Enter the Red component (0-255): </div>");

        int red = scanner.nextInt();

        System.out.print("<div data-bbox='\">Enter the Green component (0-255): </div>");

        int green = scanner.nextInt();

        System.out.print("<div data-bbox='\">Enter the Blue component (0-255): </div>");

        int blue = scanner.nextInt();

        // Check if the RGB values are within the specified range

        boolean isValidRed = red >= minRange && red <= maxRange;

        boolean isValidGreen = green >= minRange && green <= maxRange;

        boolean isValidBlue = blue >= minRange && blue <= maxRange;

        // Display results

        System.out.println("<div data-bbox='\">\nColor Component Validity:</div>");

        System.out.println("<div data-bbox='\">Red: <div data-bbox='\"> + (isValidRed ? <div data-bbox='\">Valid<div data-bbox='\"> : <div data-bbox='\">Invalid<div data-bbox='\">));

        System.out.println("<div data-bbox='\">Green: <div data-bbox='\"> + (isValidGreen ? <div data-bbox='\">Valid<div data-bbox='\"> : <div data-bbox='\">Invalid<div data-bbox='\">));

        System.out.println("<div data-bbox='\">Blue: <div data-bbox='\"> + (isValidBlue ? <div data-bbox='\">Valid<div data-bbox='\"> : <div data-bbox='\">Invalid<div data-bbox='\">));

        // Close the scanner

        scanner.close();

    }

}
```

## OUTPUT :

```
Enter the Red component (0-255): 235
Enter the Green component (0-255): 125
Enter the Blue component (0-255): 255

Color Component Validity:
Red: Valid
Green: Valid
Blue: Valid
```

2. To build a TrafficLightChecker class, you should focus on creating a system that simulates the behavior of traffic lights. This simulation can be expanded to include functionalities such as checking the current light

status, determining the duration of each light phase, and providing a mechanism for switching between lights.

```
import java.util.Scanner;

public class TrafficLightChecker {
    // Enum to define traffic light states
    private enum TrafficLight {
        RED, YELLOW, GREEN
    }

    // Method to get the next traffic light based on current light
    private static TrafficLight getNextLight(TrafficLight current) {
        switch (current) {
            case RED:
                return TrafficLight.GREEN;
            case YELLOW:
                return TrafficLight.RED;
            case GREEN:
                return TrafficLight.YELLOW;
            default:
                throw new IllegalArgumentException("&quot;Unexpected value: &quot; +
                    current);
        }
    }
}
```

```

}

// Method to display the traffic light status
private static void displayStatus(TrafficLight light) {
    switch (light) {
        case RED:
            System.out.println("The light is RED. Please stop.");
            break;
        case YELLOW:
            System.out.println("The light is YELLOW. Prepare to stop.");
            break;

        case GREEN:
            System.out.println("The light is GREEN. You may go.");
            break;
    }
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    // Prompt user for the initial traffic light state
    System.out.print("Enter the current traffic light color (RED, YELLOW, GREEN): ");
    String input = scanner.next().toUpperCase();
    TrafficLight currentLight;
    try {
        // Convert the input string to TrafficLight enum
        currentLight = TrafficLight.valueOf(input);
    } catch (IllegalArgumentException e) {
        System.out.println("Invalid color entered. Please enter RED, YELLOW, or GREEN.");
    }
    scanner.close();
    return;
}

```

```

// Display the current light status
displayStatus(currentLight);

// Determine the next traffic light state
TrafficLight nextLight = getNextLight(currentLight);

// Display the next light status
System.out.println("The next light will be: " + nextLight);
displayStatus(nextLight);


// Close the scanner
scanner.close();
}
}

```

### OUTPUT :

```

Enter the current traffic light color (RED, YELLOW, GREEN): RED
The light is RED. Please stop.
The next light will be: GREEN
The light is GREEN. You may go.

```

3. To implement a TrafficLightSwitch class that simulates switching traffic lights, you might want to create functionality for managing the current state of the traffic light, switching between states, and possibly displaying information about the light. Below is a detailed example of how you could set up this class:

```

import java.util.Scanner;

public class TrafficLightSwitch {

    // Enum to define traffic light states
    private enum TrafficLight {
        RED, YELLOW, GREEN
    }

    // Method to get the next traffic light based on current light
    private static TrafficLight getNextLight(TrafficLight current) {
        switch (current) {
            case RED:
                return TrafficLight.GREEN;
            case YELLOW:

```

```

return TrafficLight.RED;
case GREEN:
return TrafficLight.YELLOW;
default:
throw new IllegalArgumentException(""Unexpected value: " + current);
}
}

// Method to display the traffic light status
private static void displayStatus(TrafficLight light) {
switch (light) {
case RED:
System.out.println(""The light is RED. Please stop.");
break;

case YELLOW:
System.out.println(""The light is YELLOW. Prepare to stop.");
break;
case GREEN:
System.out.println(""The light is GREEN. You may go.");
break;
}
}

public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
// Prompt user for the initial traffic light state
System.out.print(""Enter the current traffic light color (RED, YELLOW,
GREEN): ");
String input = scanner.next().toUpperCase();
TrafficLight currentLight;
try {
// Convert the input string to TrafficLight enum
currentLight = TrafficLight.valueOf(input);

```

```

    } catch (IllegalArgumentException e) {
        System.out.println("&quot;Invalid color entered. Please enter RED, YELLOW, or
        GREEN.&quot;);
        scanner.close();
        return;
    }
    // Display the current light status
    displayStatus(currentLight);
    // Determine the next traffic light state
    TrafficLight nextLight = getNextLight(currentLight);
    // Display the next light status
    System.out.println("&quot;The next light will be: &quot; + nextLight);
    displayStatus(nextLight);
    // Close the scanner

    scanner.close();
}
}

```

## OUTPUT :

```

Enter the current traffic light color (RED, YELLOW, GREEN): YELLOW
Initial Traffic Light Status:
The light is YELLOW. Prepare to stop.
Press Enter to switch to the next light (or type 'exit' to quit):
The light is RED. Please stop.
GREEN
The light is GREEN. You may go.
RED
The light is YELLOW. Prepare to stop.
YELLOW
The light is RED. Please stop.
EXIT
Exiting Traffic Light Simulator.
}
...Program finished with exit code 0
Press ENTER to exit console.

```