String in Switch Case in Java

The switch statement is a multi-way branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression. Basically, the expression can be a byte, short, char, and int primitive data types. Beginning with JDK7, it also works with enumerated types (Enums in java), the String class, and Wrapper classes.

Hence the concept of string in switch statement arises into play in JDK 7 as we can use a string literal or constant to control a switch statement, which is not possible in C/C++. Using a string-based switch is an improvement over using the equivalent sequence of if/else statements. We now declare a string as a String class object only as depicted below:

Illustration:

String geeks = "GeeksforGeeks" ; // Valid from JDK7 and onwards

Object geeks = "GeeksforGeeks" ; // Invalid from JDK7 and onwards

There are certain key points that are needed to be remembered while using switch statement as it does provide convenience but at the same time acts as a double sword, hence we better go through traits as listed:

1. Expensive operation: Switching on strings can be more expensive in terms of execution than switching on primitive data types. Therefore, it is best to switch on strings only in cases in which the controlling data is already in string form.

2. String should not be NULL: Ensure that the expression in any switch statement is not null while working with strings to prevent a NullPointerException from being thrown at run-time.

3. Case Sensitive Comparison: The switch statement compares the String object in its expression with the expressions associated with each case label as if it were using the *equals() method* of String class consequently, the comparison of String objects in switch statements is case sensitive.

4. Better than if-else: The Java compiler generates generally more efficient bytecode from switch statements that use String objects than from chained if-then-else statements.

Example 1:

```java
// Java Program to Demonstrate use of String to
// Control a Switch Statement

// Main class
public class GFG {

    // Main driver method
    public static void main(String[] args)
    {

        // Custom input string
        String str = "two";
        // Switch statement over above string
        switch (str) {

        // Case 1
        case "one":

            // Print statement corresponding case
            System.out.println("one");

            // break keyword terminates the
            // code execution here itself
            break;

        // Case 2
        case "two":

            // Print statement corresponding case
            System.out.println("two");
            break;

        // Case 3
        case "three":
            // Print statement corresponding case
            System.out.println("three");
            break;

        // Case 4
        // Default case
        default:

            // Print statement corresponding case
            System.out.println("no match");
        }
    }
}
```

Output

two

Example 2:

```
// Java Program to Demonstrate use of String to
// Control a Switch Statement

// Main class
```

```
public class GFG {

    // Main driver method
    public static void main(String[] args)
    {

        // Custom input string
        // Null string is passed
        String str = "";

        // Switch statement over above string
        switch (str) {

        // Case 1
        case "one":

            // Print statement corresponding case
            System.out.println("one");

            // break keyword terminates the
            // code execution here itself
            break;
        // Case 2
        case "two":

            // Print statement corresponding case
            System.out.println("two");
            break;

        // Case 3
        case "three":

            // Print statement corresponding case
            System.out.println("three");
            break;

        // Case 4
        // Default case
        default:

            // Print statement corresponding case
            System.out.println("no match");
        }
```

```
       }
}

```

Output

```
no match
```