

```
switch:  
-----  
'if' is able to perform single condition checking, but, switch is able to perform  
Multiple Conditions checking.  
  
Note: if-elseif-else syntax is able to represent switch.  
  
In general, we will use switch in menu driven applications.
```

## Syntax:

```
-----  
switch(val)  
{  
    case 1:  
        ---instructions---  
    break;  
    case 2:  
        ---instructions---  
    break;  
    -----  
    -----  
    case n:  
        ---instructions---  
    break;  
}
```

```
class Test
{
    public static void main(String[] args)
    {
        int a = 5;           I
        switch (a)
        {
            case 1:
                System.out.println("Monday");
            break;
            case 2:
                System.out.println("Tuesday");
            break;
            case 3:
                System.out.println("Wensday");

        }
    }
}
```

```
case 4:
    System.out.println("Thursday");
break;
case 5:
    System.out.println("Friday");
break;
case 6:
    System.out.println("Saturday"); I
break;
case 7:
    System.out.println("Sunday");
break;
default:
    System.out.println("Provide Value from 1 to 7");|
```

```
        break;
    case 5:
        System.out.println("Friday");
    break;
    case 6:
        System.out.println("Saturday");
    break;
    case 7:
        System.out.println("Sunday"); I
    break;
    default:
        System.out.println("Provide Value from 1 to 7");
    break;
}

}
```

```
1 class Test
2 {
3     public static void main(String[] args)
4     {
5         int a = 10;
6         switch (a)
7         {
8             case 5:
9                 System.out.println("Five");
10            break;
11            case 10:
12                System.out.println("Ten");
13                break;
14            case 15:
15                System.out.println("|
16
17        }
18    }
```

```
case 20:
    System.out.println("Twenty");
break;
default:
    System.out.println("default");
break;|
```

```
{  
    char c = 'A';  
    switch (c)  
    {  
        case 'A':  
            System.out.println("AAA");  
        break; |  
        case 'B':  
            System.out.println("BBB");  
        break;  
        case 'C':  
            System.out.println("CCC");  
        break;  
        case 'D':  
            System.out.println("DDD");  
        break;  
        default:  
            System.out.println("default");  
    }  
}
```

```
{  
    boolean b = true;  
    switch (b)  
    {  
        case true:  
            System.out.println("True");  
        break;  
        case false:  
            System.out.println("False");  
        break;  
  
        default:  
            System.out.println("default");  
        break;  
    }  
}
```

```
D:\core_java6>javac Test.java  
Test.java:6: error: incompatible types: boolean cannot be converted to int  
        switch (b)  
                  ^  
1 error
```

```
1 {
2     String str = "BBB";
3     switch (str)
4     {
5         case "AAA":
6             System.out.println("AAA");
7             break;
8         case "BBB":
9             System.out.println("BBB");
10            break;
11        default:
12            System.out.println("default");
13            break;
14    }
15 }
```

```
4 {
5     String str = "ZZZ";
6     switch (str)
7     {
8         case "AAA":
9             System.out.println("AAA");
10            break;
11        case "BBB":
12            System.out.println("BBB");
13            break;
14    }
15 }
16 }
```

File Edit Format View Help

Rules and Regulations to use switch in JAVA:

---

1. switch is able to allow the data types like byte, short, int and char as parameter. Switch is not allowing the data types like long, float ,double and boolean as parameter.

2. Upto JAVA6 version, switch is not allowing String data type as parameter, but, JAVA7 version is allowing String data type as parameter.

3. In switch, both cases and default are optional, we can write switch with cases and with out default, we can write switch with out cases and with default, we can write switch with out both cases and default, but, it is not suggestible in application development.

```
int a = 10;
switch(a)
{
    case 5:
        System.out.println("Five");
    case 10:
        System.out.println("Ten");
    case 15:
        System.out.println("Fifteen");
    case 20:
        System.out.println("Twenty");
    default:
        System.out.println("Default");
```

```
D:\core_java6>java
Ten
Fifteen
Twenty
Default
```

```
D:\core_java6>
```

4. In switch, break statement is optional, if we provide switch with out break statements then JVM will execute all the cases right from matched case until it encounter either end of switch or break.

```
5     byte b = 126;
6     switch(b)
7     {
8         case 125:
9             System.out.println("125");
10            break;
11        case 126:
12            System.out.println("126");
13            break;
14        case 127:
15            System.out.println("127");
16            break;
17        case 128:
18            System.out.println("128");
19            break;
20        default:
21            System.out.println("Default");
22            break;
23    }
```

5. In switch, all case values must be provided with in the range of the data type which we provided as parameter to switch.

```
{  
    int i = 5, j = 10, k = 15, l = 20;  
    switch(10)  
    {  
        case i:  
            System.out.println("Five");  
            break;  
        case j:  
            System.out.println("Ten");  
            break;  
        case k:  
            System.out.println("Fifteen");  
            break;  
        case l:  
            System.out.println("Twenty");  
            break;  
        default:  
            System.out.println("Default");  
    }
```

```
D:\core_java6>javac Test.java
Test.java:8: error: constant expression required
        case i:
                  ^
Test.java:11: error: constant expression required
        case j:
                  ^
Test.java:14: error: constant expression required
        case k:
                  ^
Test.java:17: error: constant expression required
        case l:
                  ^
```

5. In switch, all case values must be provided with in the range of the data type which we provided as parameter to switch.

6. In switch, all case values must be constants only including final variables, not possible to provide non final variables.