

## Decisions in Java – Switch Statement

### Several Actions – The Switch Statement

In addition to the *nested if statement*, Java provides a second method for choosing from many alternative actions. Compare the following code, each of which accomplishes the same task.

#### Nested If Statement

```
if (grade == 'A')
{
    System.out.println("Excellent!");
}
else if (grade == 'B')
{
    System.out.println("Good!");
}
else if (grade == 'C')
{
    System.out.println("Average.");
}
else if (grade == 'D')
{
    System.out.println("Poor.");
}
else if (grade == 'F')
{
    System.out.println("Need help!");
}
else
{
    System.out.println("Invalid grade.");
}
```

#### Swtich Statement

```
switch (grade)
{
    case 'A':
        System.out.println("Excellent!");
        break;
    case 'B':
        System.out.println("Good!");
        break;
    case 'C':
        System.out.println("Average.");
        break;
    case 'D':
        System.out.println("Poor.");
        break;
    case 'F':
        System.out.println("Need help!");
        break;
    default:
        System.out.println("Invalid
grade.");
        break;
}
```

The `default` case, at the end of the `switch` statement, acts similarly to the `else` at the end of the nested `if`. It will catch any value that doesn't have an exact match in the cases. Although it is not necessary to include the `default` case (just as it is not necessary to include the `else`), it is good programming practise to account for any unexpected values.

One of the significant limitations of the `switch` statement is the expression that can be used to control it. With an `if` statement, anything can be compared to produce an true or false result (e.g., primitive data such as `int` or `float`, more complex data such as `Strings`, or even objects created by the user).

In a `switch` statement, however, the expression controlling the `switch` must have an integer representation, which limits it to the following data types: `byte`, `short`, `int`, `long`, or `char`.

It is also possible to combine cases so they execute the same code, as shown in the following example.

## Decisions in Java – Switch Statement

Example 1 – This switch statement assigns a grade based on a quiz that was scored out of five.

```
switch (score)
{
    case 5:
        grade = 'A';
        break;
    case 4:
        grade = 'B';
        break;
    case 3:
        grade = 'C';
        break;
    case 2:
    case 1:
    case 0:
        grade = 'F';
        break;
    default:
        System.out.println("Invalid score - grade of ? assigned");
        grade = '?';
        break;
}
```

### Exercises

1. Write a program that reads the month as a number (from 1 to 12) and then prints the month as a string. For example, input of **1** would result in the output of **January**.
2. A sequence of six tests, all scored out of 100, are to be given different weightings in determining a final mark. Write a program fragment that computes the appropriate weighted score for one test. The fragment should first read values of **testNumber** and **score**. Using a **switch** statement, it should compute and print the appropriate value of **weightedScore** using the weightings given in the following table.

Test Number	Weight
1	10%
2	20%
3	20%
4	15%
5	15%
6	20%

For example, input of **3** and **27** should produce the following output:

**A score of 27 on test 3 gives a weighted score of 5.4**

## Decisions in Java – Switch Statement

### Solutions

```
1. class PrintMonth
{
    public static void main (String [] args)
    {
        int monthNumber;

        System.out.println("Enter a number from 1 to 12");
        monthNumber = In.getInt();

        switch (monthNumber)
        {
            case 1:
                System.out.println("January");
                break;
            case 2:
                System.out.println("February");
                break;
            case 3:
                System.out.println("March");
                break;
            case 4:
                System.out.println("April");
                break;
            case 5:
                System.out.println("May");
                break;
            case 6:
                System.out.println("June");
                break;
            case 7:
                System.out.println("July");
                break;
            case 8:
                System.out.println("August");
                break;
            case 9:
                System.out.println("September");
                break;
            case 10:
                System.out.println("October");
                break;
            case 11:
                System.out.println("November");
                break;
            case 12:
                System.out.println("December");
                break;
            default:
                System.out.println("Not a month!");
                break;
        }
    }
}
```

## Decisions in Java – Switch Statement

```
2. System.out.println("Test number?");
   testNumber = In.getInt();

   System.out.println("Score?");
   score = In.getFloat();

   switch(testNumber)
   {
       case 1:
           weightedScore = 0.10 * score;
           break;
       case 2:
       case 3:
       case 6:
           weightedScore = 0.20 * score;
           break;
       case 4:
       case 5:
           weightedScore = 0.15 * score;
           break;
       default:
           System.out.println(testNumber + " is not a valid test number.");
           break;
   }

   System.out.println("A score of " + score + "on test " + testNumber +
       "gives a weighted score of " + weightedScore);
```