Lesson 14: The *switch* Statement and *char*

The *if* statement is the most powerful and often used decision-type command. The *switch* statement is useful when we have an integer variable that can be one of several quantities. For example, consider the following menu scenario (enter and run this program):

```
//This code should be placed inside the main method of a class
System.out.println("Make your arithmetic selection from the choices below:\n");
System.out.println(" 1. Addition");
System.out.println(" 2. Subtraction");
System.out.println(" 3. Multiplication");
System.out.println(" 4. Division\n");
System.out.print(" Your choice? ");

Scanner kbReader = new Scanner(System.in);
int choice = kbReader.nextInt( );

System.out.print("\nEnter first operand. " );
double op1 = kbReader.nextDouble( );

System.out.print("\nEnter second operand." );
double op2 = kbReader.nextDouble( );

System.out.println("");

switch (choice)
{
        case 1: //addition
                System.out.println(op1 + " plus " + op2 + " = " + (op1 + op2) );
                break;
        case 2: //subtraction
                System.out.println(op1 + " minus " + op2 + " = " + (op1 - op2) );
                break;
        case 3: //multiplication
                System.out.println(op1 + " times " + op2 + " = " + (op1 * op2) );
                break;
        case 4: //division
                System.out.println(op1 + " divided by " + op2 + " = " + (op1 / op2) );
                break;
        default:
                System.out.println("Hey dummy, enter only a 1, 2, 3, or 4!");
}
```

The optional *default*:

The *default* command is optional. You can use it if there might be a possibility of the value of *choice* not being one of the cases.

Give me a *break*:

The *break* statements are normally used. Try leaving them out and see what happens here. In the next section we will look at an application in which they are omitted.

Basically, *break* jumps us out of the *switch* structure and then code execution continues with the first line immediately after the closing *switch* brace. Specifically, you might want to omit the *break* within the *case 1:* section. If *choice* is 1 then the result will be that it prints the answer for **both** addition and subtraction.

The next experiment you might want to do is to leave the parenthesis off of *(op1 + op2)* in the *case 1:* section. Since *op1 + op2* is no longer in parenthesis, the plus between them no longer means addition. It now means concatenation since all the activity to the left of this point in the code was also *String* concatenation.

Leaving off the *break*:

Now, let's look at an example where we intentionally omit *break*:

```
//Suppose at this point in the program we have an integer variable, j. If j equals 1,
//2, or 3 we want to set String variable s to "low" and if j equals 4, 5, or 6 we want
//to set s to "high". If j equals 7, set s to "lucky".
        switch ( j )
        {
                case 1:
                case 2:
                case 3:
                        s = "low";
                        break;
                case 4:
                case 5:
                case 6:
                        s = "high";
                        break;
                case 7:
                        s = "lucky";
        }
```

A new data type… *char*:

Before we look further at the *switch* statement, we must look at a new data type, *char*. This stands for character. Following is a typical way to declare and initialize a character:

```
char ch = 'h';
```

Notice that a character is always enclosed in single quotes. Characters can be anything, even numbers or symbols:

```
char x = '6'; char pp = '@';
```

***int* and *char* are permissible types:**

switch( ) statements can switch on both **integers** and **characters** (*short* and *byte* types can also be used, but rarely are). Modify the example on the previous page to switch on a *char* instead of *int*. See the next page for the necessary modifications:

```
System.out.println("Make your arithmetic selection from the choices below:\n");
System.out.println(" A. Addition");
System.out.println(" S. Subtraction");
System.out.println(" M. Multiplication");
System.out.println(" D. Division\n");
System.out.print(" Your choice? ");

Scanner kbReader = new Scanner(System.in);
String choice = kbReader.nextLine( );

//char ch = choice; //You would think this would work…but it doesn't.
char ch = choice.charAt(0); //you just learned another String method.

System.out.print("\nEnter first operand. " );
double op1 = kbReader.nextDouble( );

System.out.print("\nEnter second operand ." );
double op2 = kbReader.nextDouble( );

System.out.println(" ");

switch (ch)
{
        case 'A': //addition
        case 'a': //Notice we are providing for both capital A and little a.
                System.out.println(op1 + " plus " + op2 + " = " + (op1 + op2) );
                break;
        case 'S': //subtraction
        case 's':
                System.out.println(op1 + " minus " + op2 + " = " + (op1 - op2) );
                break;
        case 'M': //multiplication
        case 'm':
                System.out.println(op1 + " times " + op2 + " = " + (op1 * op2) );
                break;
        case 'D': //division
```

```
        case ‘d’:
                System.out.println(op1 + “ divided by ” + op2 + “ = ” + (op1 / op2) );
                break;
                default:
                System.out.println(“Hey dummy, enter only a A, S, M, or D!”);
}
```

With the advent of Java 7.0, it is also possible to switch on *String*s as demonstrated below:

```
        String s = “Hello”;
        switch(s)
        {
                case “Good Bye”:
                        System.out.println(“You said Good Bye.”);
                        break;
                case “Hello”:
                        System.out.println(“You said Hello.”);
                        break;
                case “Auf wiedersehen”:
                        System.out.println(“You spoke German.”);
                        break;
                default:
                        System.out.println(“What did you say?”);
        }
```