Switch statement

The switch statement provides an alternative to the if-statement. It is usually the better alternative when there is a choice among several values.

The syntax of the switch statement is

switch (<expression>) <block>

The <block> is a sequence of statements interspersed with “switch labels” of the form:

case <value> :

and there may be one default switch label:

default:

```
/* Store college-year year as a
* String in y.
* If year is not in 1..4, store the
* empty string in String y; */
switch (year) {
   case 1: y= "freshman"; break;
   case 2: y= "sophomore"; break;
   case 3: y= "junior"; break;
   case 4: y= "senior"; break;
   default: y= "";
}
```

The switch labels may be in any order —even the default switch label could be first. The <expression> and the <values> must have the same type. More on possible types later. In the switch statement in the box above, variable year is of type int; it could be any expression of type int.

Consider executing the switch statement in the box above, with int variable year containing 2. Then, the statements following the switch label “case 2” are executed. Thus, "sophomore" is stored in y and then execution of the break statement terminates execution of the switch[1]. Similarly for year being 1, 3, and 4.

But if year contains any other value, the statements following the default switch label are executed, so the empty string is stored in y.

General execution of the switch statement

Here is a more general explanation of execution of a switch statement: Evaluate the expression to get some value v. Three possible cases arise:

1. There is a switch label “case v”. Execute the statements in the <block> beginning with the first one following that switch label.
2. There is no switch label “case v” and there is a default switch label. Execute the statements in the <block> beginning with the first one following the default switch label.
3. There is no switch label “case v” and there is no default switch label. Execution of the switch is finished.

Importance of the break statement

The break statement should be used to terminate execution of the switch statement where necessary. It was used in the box above.

Look at the box to the right. First, notice that the switch <expression> is a String and that each of the switch labels has a String value.

Second, there are no break statements. Thus, for the case that y contains "freshman", all three println statement are executed! This is desired in this case, since, based on the specification, if y contain "freshman", "sophomore", "junior", and "senior" should be printed.

```
// Print college years following year y
switch (y) {
   case "freshman":
      System.out.println("sophomore");
   case "sophomore":
      System.out.println("junior");
   case "junior":
      System.out.println("senior");
   case "senior":
}
```

Third, note that there is no default switch label. Therefore, if y contains "frosh", no statements are executed and the switch statement terminates without printing anything.

[1] Terminating execution of the switch statement means just that: its execution is finished, done. If there is another statement following it, that statement will be executed next. But in explaining execution of the switch, or any statement, we do nothing more than explain its execution and not what happens before or after it.

Switch statement

Possible types of the <expression>

The <expression> in a switch statement can have any of these types:

byte, short, char, and int
wrapper classes Byte, Short, Character, and Integer
String
any enum type

The types of values in the switch labels must be the same as the type of the <expression>, although autoboxing and unboxing will be done where possible.

Example using type char

The box to the right gives an example of a switch statement that switches on a value of type char. It also shows the indentation that is normally used for switch statements —use the automatic indentation feature provided by Eclipse and DrJava and it will format the lines as shown.

The example also shows two switch labels next to each other, something that did not occur in the examples given on the previous page.

Note that no break statement is needed after the println statement in case 'D', although many people will put it there just for consistency

```
/* print "Excellent" if grade is 'A',
 * "OK" if grade is 'B' or 'C',
 * "Passing" if grade is 'D',
 * "Fail" if grade is 'F'. */
switch (grade) {
   case 'A':
      System.out.println("Excellent");
      break;
   case 'B': case 'C':
      System.out.println("OK");
      break;
   case 'D':
      System.out.println("Fail");
}
```