Lab 3

Using Operators and Decision Constructs

Objectives

Upon completion of this lab, you should be able to:

- Complete review questions
- Use `if` and `if/else` constructs in a Java program
- Use `switch` construct in a Java program

Lab Overview

In this lab, you complete the review questions and two exercises.

- In the first exercise, you create if and if/else constructs.
- In the second exercise, you create `switch` constructs

Completing Review Questions

Complete the following questions:

1. What is the purpose of the else block in an if-else statement?

   a. To contain the remainder of the code for a method.

   b. To contain code that is executed when the expression in an `if` statement is `false`.

   c. To test whether an expression is `false`.

2. Which of the following sentences is suitable for testing a value in a `switch` construct?

   a. The `switch` construct tests whether values are greater than or less than a single value.

   b. The `switch` construct tests against a single variable.

   c. The `switch` construct tests the value of a float, double, or boolean data type.

3. What relational operators and conditional operators are mentioned in the following paragraph?
   If the time is 8:00 a.m., then it is time for work and time for coffee. If the time is greater than 12:00 p.m., it is time for lunch or time for tea. If the time is 5:00 p.m. or later, it is time to go home.

   a. ==, >, =>

   b. ==, &&, <

   c. ==, &&, >

   d. ==, `||`, >

4. Read the following code snippet and identify the correct options:

```
1 String clothing1;
2 String clothing2;
3 clothing1 = "socks";
4 clothing2 = "pants";
5 if (clothing1 == clothing2)
```

   a. Line 5 tests if the String literals pointed to by the `clothing1` and `clothing2` references are equal

   b. Line 5 tests the String objects' references in memory, not their contents.

   c. By replacing Line 5 with the `if (clothing1.equals(clothing2))` statement, the string literals referenced by `clothing1` and `clothing2` are equal.

   d. Line 5 always returns `true`

Exercise 1: Using if and if/else Constructs

The objective of this exercise is to create classes that use if and if/else constructs.

Preparation

Ensure that DateTwoTest.java and ClockTest.java files exists in the SL110/exercises/06_decision/exercise1 directory. This is your working directory.

This exercise has two tasks. In each task you create a class and use the if/else statements where ever applicable. The tasks are:

- "Task 1 – Writing a Class That Uses if/else Statements"
- "Task 2 – Writing Another Class That Uses if/else Statements"

Task 1 – Writing a Class That Uses if/else Statements

In this task, you write a class called DateTwo that uses if/else statements to display the day of the week based on the value of a variable. Follow these steps to write your class:

1. Go to your working directory. Create a class called DateTwo with one member variable called dayNumber. Assign a value to dayNumber between 1 and 7 where:
   - The number 1 represents Monday (beginning of the week).
   - The number 7 represents Sunday (end of the week).
2. In the DateTwo class, create a displayDay method that uses if/else constructs to inspect the value of the dayNumber and displays the corresponding day of the week. The displayDay method should also display an error message if an invalid number is found.
3. Save, compile, and execute your class using the provided DateTwoTest class file.
4. Repeat Step 3 several times by assigning different values to the DateTwo member variable dayNumber in the DateTwoTest.java file.

Task 2 – Writing Another Class That Uses `if/else` Statements

In this task, you write a class called `Clock` that uses `if/else` statements to display the part of day depending on the time of day. Use the following table as a guideline.

Time of Day	**Part of Day**
8:01 to 12:00	Morning
12:01 to 17:00	Afternoon
17:01 to 24:00	Evening
0:01 to 8:00	Early Morning

Follow these steps to write your class:

1. Go to your working directory.
2. Create a class called `Clock` with a variable called `currentTime` that contains the hour of day.
3. In the `Clock` class, create a `displayPartOfDay` method that uses `if/else` constructs to display the part of the day associated with the value of the `currentTime` variable. For example, if the value of the `currentTime` variable is equal to 1504, your program would display "Afternoon."
4. Compile and execute your program using the `ClockTest` class file.
5. Repeat Step 4 several times by assigning different values to the Clock member variable `currentTime` in the `ClockTest.java` file.

Note – A leading zero indicates an octal value. Therefore, the program does not compile if you set `currentTime` to 0800. You need to specify `currentTime` as 800 for 8:00 AM to successfully compile the program. No tests have been done for values that lie outside the range of 100 and 2400.

Exercise 2: Using the Switch Statement

The objective of this exercise is to practice using the switch construct in decision-making programs. In this exercise, you create a program that displays the name of the month based on the month number.

Preparation

Ensure that the MonthTest.java file exists in the SL110/exercises/06_decision/exercise2 directory. The is your working directory.

Task – Writing a class That Uses the switch Statement

In this task you create a class called Month that uses switch statements to display the name of the month based on the numeric value of a variable.

Complete the following steps

1. In your working directory, create a class called Month and save the file as Month.java.
2. In the Month class, create a variable called monthNumber. Assign a value to the variable that is between 1 and 12, where the number 1 represents the month of January and the number 12 represents the month of December.
3. In the Month class, create a displayMonth method that uses a switch construct to inspect the value of the monthNumber variable and displays the corresponding name of the month. The displayMonth method should also display an error message if an invalid number is used.
4. Save and compile the file.
5. Execute the program by running the MonthTest class file.
6. Repeat Step 5 several times assigning different values to the Month member variable monthNumber in the MonthTest.java file.

Exercise Summary

Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications