Lab 3

Developing and Testing a Java Technology Program

Objectives

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

- Complete review questions
- Write, modify, compile, and execute Java technology classes

Lab Overview

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

- In the first exercise, you compile and execute an existing Java technology class.
- In the second exercise, you write, compile, and execute your first Java technology class.

Completing Review Questions

Complete the following questions:

1. Which of the following options is part of the class declaration syntax:
   a. Class arguments
   b. Method identifier
   c. Modifier
   d. `//`
2. Identify the statement below that best describes the main method:
   a. The `main` method is the starting point for all Java technology applets.
   b. The `main` method usually has a return type of `int`.
   c. The `main` method is required in all Java classes.
   d. The `main` method is a special method that the Java Virtual Machine recognizes as the starting point for every Java technology program run from a command line or a prompt.
3. State whether the following statements are true or false:
   a. The `java` command executes a class or bytecode file.
   b. Every open curly bracket in a Java technology program must have a corresponding closing curly brace.
   c. `*/` is a valid starting delimiter for a comment.
   d. The `javac` command reads class and interface definitions, written in the Java programming language, and compiles them into bytecode class files.

Exercise 1: Modifying, Compiling, and Executing a Java Program

A Java technology program is already created for you. You need to open it, examine the lines of code, modify it, compile it, and then test it by executing the program.

Preparation

Ensure that `QuotationTest.java` and `Quotation.java` files exist in `SL110/exercises/03_getstarted/exercise1` directory

.Task – Compiling and Executing a Java Program

In this task, you will modify a Java Program, save, compile and test it.

Complete the following steps:

1. Open a text editor.
2. Open the Quotation.java file located at `SL110/exercises/03_getstarted/exercise1` directory.
3. Examine the Quotation class and identify its member variable.
4. In the `display` method, write code to display the value of the member variable.
5. Save and close the `Quotation.java` file.
6. Open the `QuotationTest.java` file located at `SL110/exercises/03_getstarted/exercise1` directory and examine its `main` method. The main method creates an instance of the Quotation class and executes the `display` method of `Quotation`. Close the file.
7. Open the terminal window and navigate to the directory `SL110/exercises/03_getstarted/exercise1`.
8. Compile `Quotation.java` file using `javac`. Foe example: **`javac Quotation.java`**

9. If required, modify the `Quotation.java` file to correct any compilation errors generated by the previous step. Save the file and recompile it.
10. Compile `QuotationTest.java` file using `javac`. For example: **`javac QuotationTest.java`**
11. Examine the `SL110/exercises/03_getstarted/exercise1` directory to verify if the compilation steps has created the following class files:
    - `Quotation.class`
    - `QuotationTest.class.`
12. Execute `QuotationTest.class` file using java and verify the output. For example: **`java QuotationTest`**
13. Open `Quotation.java` file, modify the default value of the member variable in `Quotation.java` file to your own favourite quotation, and save the file.
14. Compile `Quotation.java` file.
15. Execute `QuotationTest.class` file and verify the changed output.

Exercise 2: Creating, Compiling, and Executing a Java Class

In this exercise you will create a Java class, and compile it. You will also create another Java class to test the previous class.

Task – Creating a Java Class and Testing It

In this task, you will create two classes and compile them. You will execute one class file.

Complete the following steps:

1. Go to the `SL110/exercises/03_getstarted/exercise2` directory.
2. Open an editor, and enter in the Java technology syntax for the `Shirt` class shown in Code 3- 1 of this lab.

Code 3-1 `Shirt.java`

```
1 public class Shirt {
2
3 public int shirtID = 0; // Default ID for the shirt
4 public String description = “description required-”;
// default
5 // The color codes are R=Red, B=Blue, G=Green,
U=Unset
6 public char colorCode = ‘U’;
7 public double price = 0.0; // Default price for all
shirts
8 public int quantityInStock = 0; // Default quantity
for all shirts
9
10 // This method displays the values for an item
11 public void displayShirtInformation() {
12 System.out.println(“Shirt ID: “+ shirtID);
13 System.out.println(“Shirt description:” +
description);
14 System.out.println(“Color Code: “+ colorCode);
15 System.out.println(“Shirt price: “+ price);
16 System.out.println(“Quantity in stock: “+
quantityInStock);
18 } // end of display method
19 } // end of class
```

3. Save and close the file.
4. Open an editor, and enter the `ShirtTest` class shown in Code 3-2 of this lab.

Code 3-2 `ShirtTest.java`

```
1 public class ShirtTest {
2
3 public static void main (String args[]) {
4
5 Shirt myShirt;
6 myShirt = new Shirt();
7
8 myShirt.displayShirtInformation();
9
10
11 }
12
```

5. Save and close the file.
6. Open a terminal window and navigate to the directory for this module.
7. Type the command to compile the `Shirt` class into an executable bytecode file. For example:
   `javac Shirt.java`
8. Verify that the file `Shirt.class` is created in the `SL110/exercises/03_getstarted/exercise2` directory.
9. Type the command to compile the `ShirtTest class` into an executable bytecode file.
   `javac ShirtTest.java`
10. Verify that the file `ShirtTest.class` is created in the `SL110/exercises/03_getstarted/exercise2` directory.
11. Type the command to run the `ShirtTest.class` file and examine the output of the program.
    `java ShirtTest`
12. Open the `Shirt.java` file in the editor again.
13. Modify the values of the `ShirtID` and `price` member variables.
14. Save and close the `Shirt.java` file.
15. Recompile `Shirt.java` file at the terminal window.
    `javac Shirt.java`

16. Run the ShirtTest.class file at the terminal window.
    `java Shirt`
17. Examine the output of the program and verify if the output is differs from the output as seen in Step 9.

Exercise Summary

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

- Experiences
- Interpretations
- Conclusions
- Applications