

## 05.04 Instances of a Class

Student: Trisha Ganesh

(starts on page 4)

## 05.04 Assignment Instructions

In your Language Arts classes, you have probably been asked to compare and contrast the plot, setting, characters, imagery, vocabulary, metaphors, rhetorical strategies, etc. of different types of literature (e.g., *The Color Purple* vs. *The Scarlet Letter*, *To Kill a Mockingbird* vs. *Fahrenheit 451*, or *The Crucible* vs. the *Scarlet Letter*). Your literary analysis skills can be transferred to comparing and contrasting programs written in different styles, too. You have been developing keen observational skills as your Java programming fluency has steadily increased. You are aware that programs exhibit a definite syntax and style, organizational structure, vocabulary, flow of control, etc., which can be gleaned by careful observation during a desk check.

In this assignment, you will apply some of the writing lessons you have learned about comparing and contrasting, but in a different context. The source code for three programs is provided below. Using a word processor, write a short essay (approximately 300 words) comparing and contrasting the source code of the three programs. Conclude your essay by giving your opinion about the style you currently prefer and the reason(s) for your choice.

### Program 1

```
public class HelloWorldV1

{

//main method

public static void main(String[] args)

{

System.out.println("Hello, Virtual World!");
```

```
System.out.println("It is a great day for programming.");

}

}
```

## Program 2

```
public class HelloWorldV2

{

//print two lines of text

public static void printTwoLines( )

{

System.out.println("Hello, Virtual World!");

System.out.println("It is a great day for programming.");

}

//main method

public static void main(String [] args)

{

printTwoLines();

}

}
```

## Program 3

```
public class HelloWorldV3

{

    //default constructor

    public HelloWorldV3()

    {

    }

    //print two lines of text

    public void printTwoLines( )

    {

        System.out.println("Hello, Virtual World!");

        System.out.println("It is a great day for programming.");

    }

    //main method

    public static void main(String [] args)

    {
```

```
HelloWorldV3 hello = new HelloWorldV3( );

hello.printTwoLines();

}

}
```

#### Updated:

Overall, the first program just uses a single main method to do the work which can get complicated as the program grows. If we want to reuse the code or print the two lines more than once, we will need to cut and paste the code at various places in the main method.

The second program defines a different method to print which is good and will allow us to call it multiple times. Also divides the code into different pieces to make things simpler.

Program 3 is object-oriented which is what Java is about. It uses classes and objects to help organize the code and possibly make programs easier to write as they get more complex.

For this course, I've mostly succeeded using methods 1 and 2. So, I am currently more comfortable with 1 and 2. However, with more practice, I will be able to utilize the concepts presented in program 3.

#### Previously submitted:

These programs consist of various concepts taught in this course. The first program has two print statements in the main class. When you run this program, the output will be the following:

Hello, Virtual World!

It is a great day for programming.

In the second program, you are also trying to essentially print the same two lines. However, the only difference is that you are doing what is called "calling a method". This will get it to perform the task. For example, public static void printTwoLines( ) at the top, is being called by the printTwoLines() right below the main method. When you call it, whatever is inside the public static void printTwoLines( ), will be executed. So, the output will be the same as the first program. The only difference is that in this program, you are calling a method. This can be seen because of the empty parentheses which indicate that we, the users, don't need to input or write anything for the output to be displayed.

In the third program, you can see at the very top, a default constructor, a constructor with no parameters, had been added. This is used to create the object "hello" in the program which can be seen at the very bottom of the program. This program is different from the first and second because of the default constructor and how it created the object hello. In addition, you'll notice that the previous program, program 2, had public static void printTwoLines( ). Program 3, on the other hand, doesn't have static. It only has void. This is because of the new "hello" object we created which is located here: `hello = new HelloWorldV3( );` You can see that the default constructor, `HelloWorldV3( )`, has been repeated to show the object "hello" being created and since a new object is being created, you add a new before the `HelloWorldV3( );`

All programs used `System.out.println()`. However, the first program was only in/using the main class/method, the second was calling a method, and the third was using/creating a default constructor to create an object which then presents the output.