HERBERT SCHILDT'S *Java: A Beginner's Guide*, 6TH ED.

Chapter 1 Self Test Answers

1. **What is bytecode and why is it important to Java's use for Internet programming?**

   Schildt states "Bytecode is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the *Java Virtual Machine (JVM)*. In essence, the first JVM was designed as an *interpreter for bytecode*." Bytecode is what is created by running the "javac" command in the terminal window, as in:

   ```
   javac MyProgram.java
   ```

   This compiles the .java file into a .class file. .class files can be run on any computer that has a JVM installed, including any computer connected to the Internet. Files are run from the terminal using is command:

   ```
   java MyProgram
   ```

   Java files can obviously be designed for and run in a GUI rather than a console application, but this is the basic way to compile and run a program in the console.

2. **What are the three main principles of object-oriented programming?**

   The three main principles of object-oriented programming are:

   (a) Polymorphism - the quality that allows one interface to access a general class of actions (p. 9).

   (b) Inheritance - the process by which one object can acquire the properties of another object (p. 10).

   (c) Encapsulation - a programming mechanism that binds together code and the data it manipulates, and that keeps both safe from outside interference and misuse (p. 9).

3. **Where do Java programs begin execution?**

   Java programs begin execution with the "main" method, the standard declaration for which is:

   ```
   public static void main (String[] args){
      // insert code into the main method here.
   }
   ```

4. **What is a variable?**

   Schildt states that "A *variable* is a named memory location that can be assigned a value. Further, the value of a variable can be changed during the execution of a program." Variables are the where programs store data and are what the methods of a program manipulate to output values.

5. **Which of the following variable names is invalid?**

   (a) `count`- valid.

   (b) `$count`- valid.

   (c) `count27`- valid.

   (d) `67count`- invalid. In Java, a variable cannot begin with a digit.

6. **How do you create a single-line comment? How do you create a multiline comment?**

   In Java, a single-line comment is created by adding two forward slashes. Multiline comments begin with a forward slash and an asterisk, and end with an asterisk and a forward slash.

   ```
   // this is a single-line comment.

   /*
   * This is a multiline comment.
   */
   ```

7. **Show the general form of the *if* statement. Show the general form of the *for* loop.**

   The general form of an *if* statement is:

```
if (x = 0){
  // This section is only executes when if statement is true
  System.out.println(”The variable ” + x + ”is equal to 0”);
}
```

   The general form of a *for* loop is:

```
for (i = 0; i < 100; i++){
  // This section will iterate 100 times.
  System.out.println(i);
}
```

8. **How do you create a block of code?**

   A block of code is created by surrounding it with curly braces. For example:

```
{
  /* This is a block of code.
  * A block of code is one single logical code statement.
  * By definition, it can be substituted for any other
  * single statement.
  */
}
```

9. **The moon's gravity is about 17 percent that of the earth's. Write a program that computes your effective weight on the moon.**

   The original file can be found in the “examplePrograms” folder.

```
class WeightOnTheMoon {

  public static void main (String[] args){

    double weight = 90;
    double weightOnMoon = weight * 0.17;

    System.out.println(”If I weigh ” + weight + ” kilograms on earth,
                      then my weight on the Moon will be ”
                      + weightOnMoon + ” kilograms.”);

  } // end main
} // end WeightOnTheMoon
```

10. **Adapt Try This 1-2 so that it prints a conversion table of inches to meters. Display 12 feet of conversions, inch by inch. Output a blank line every 12 inches (One meter equals approximately 39.37 inches).**

```
/*
*  This Question 10 from Chapter 1 Self Test
*  of Schildt's Java: A Beginner's Guide
*  This program displays a conversion table of inches to meters
*/

class InchesToMetersTable {

 public static void main(String[] args){

  double meters;
  int inches, counter;

  counter = 0;

  for (inches = 1; inches <= 144; inches ++){
   meters = inches/39.37; // convert to meters
   System.out.println(inches + ” inches is ” + meters + ” meters.”);

   counter++;
   // every 12th line, print a blank line
   if (counter == 12){
     System.out.println();
     counter = 0; // reset the line counter
   } // end if
  } // end for
 } // end main
} // end InchesToMetersTable
```

11. **If you make a typing mistake when entering your program, what sort of error will result?**

    A typing mistake while entering a program can have varying results, depending on where the error occurs. If it is within a String, the error could result in inaccurate output. More likely, however, is that the error will occur while typing syntax, which will result in a compile-time error. When attempting to compile the program, the console will produce an error message which will offer some clue as to where the error occurred, which is sometimes helpful and sometimes not.

12. **Does it matter where on a line you put your statement?**

    Java does not respect white space. This means that a line continues until a semi-colon (;) is reached. The exact placement of statements is irrelevant to Java. Neatly laid out, well-commented and visually-appealing code is only important to ease the process of a programmer reading (and understanding) code, which can be invaluable during debugging.