```
//*********************************************************************
// Programmer: Dr. Jeff Hill
// Example Assignment - 05/10/2016 - 15 minutes
//
// Description: This class demonstrates the expected coding
//  conventions including formatting, comments and naming.
//
// Integrity Statement: I pledge that this program represents my own
//  unique programming code. I further pledge that this
//  program was created specifically to satisfy the requirements
//  specified in the assignment listed above. I received help
//  from Dr. Jim Downey in designing and debugging my program.
//*********************************************************************

package class2;

public class ConventionsExample {

   // define and initialize global variables
   private static String variableOne = "Dr.";
   private static String variableTwo = "Jeff Hill";
   private static int age = 42;
   private final static int YEARGRADUATED = 1998;

   // this method is the entry point when the program is executed
   public static void main(String[] args) {
      System.out.println("My name is " + variableOne + " " + variableTwo
         + " and I am " + age + " years old. I " + " graduated in " +
         YEARGRADUATED + " with a degree in MIS.");
      System.out.println("This is a second line of text in the output.");
   } // end method main
} // end class ConventionsExample
```

Make sure to include all expected header information: Programmer, Assignment, Date, Time spent, Class Description and Integrity Statement

Proper package name

Proper class name

Proper variable names

Describe the method

Proper method names

Proper code block closing labels

Proper indentation: methods and variables “belong” to the class so are indented 1 tab-stop. Code and local variables “belong” to a method so are indented 1 additional tab-stop. Line up all closing code blocks with their opening statement.

Remove extraneous line breaks and other white space.

IDEs apply text colors as visual indicators of how it recognizes the code words

Blue = Java reserved word
Green = Global variable
Orange = Text string literal
Grey = Comment
Black = Other
Red Wavy Underline = Coding or syntax error – MUST be fixed
Yellow Wavy Underline = Compiler warning – MAY be ignored
Gray Wavy Underline = Unused variable – SHOULD be fixed