**Exercises:**

**1.2**: Which of the following are not valid Java identifiers? Why?

* + Factorial
  + anExtremelyLongIdentifierIfYouAskMe

2ndLevel *Identifiers must not start with a number*

* + level2
  + MAX\_SIZE
  + highest$

hook&ladder ***Identifiers can contain only special characters underscore \_ and dollar*** *$*

**1.3**: Why are the following valid Java identifiers not considered good identifiers?

* + q ***One character is too vague for an identifier.***
  + totVal ***The word can be spelled out completely for better clarity.***
  + theNextValueInTheList ***Long identifiers is better connected with underscores.***

**1.6**: Categorize each of the following situations as a compile-time error, run-time error, or logical error:

* + multiplying two numbers when you meant to add them ***Logical error.***
  + dividing by zero ***run-time error.***
  + forgetting a semicolon at the end of a programming statement ***compile-time error.***
  + spelling a word wrong in the output ***Logical error.***
  + producing inaccurate results ***Logical error.***
  + typing a { when you should have typed ( ***compile-time error.***

**2.2**: What output is produced by the following code fragment? Explain.

System.out.print ("Here we go!");

System.out.println ("12345");

System.out.print ("Test this if you are not sure.");

System.out.print ("Another.");

System.out.println ();

System.out.println ("All done.");

The code snippet above produces the following:

//*printed a string without new line first, followed by printing one with new line*

Here we go!12345

Test this if you are not sure.Another.

//from the new line, *print without new line twice to join 2 strings,*

//*printed nothing and terminate with a new line,*

*//printed a string [All done.] and terminated with new line*

All done.

**2.3**: What is wrong with the following program statement? How can it be fixed?

System.out.println ("To be or not to be, that is the

question.");

**2.5**: What output is produced by the following statement? Explain.

System.out.println ("He thrusts his fists\n\tagainst" +

" the post\nand still insists\n\the sees the \"ghost\"");

**2.11**: For each of the following expressions, indicate the order in which the operators will be evaluated by writing a number beneath each operator.

d. a + b / c \* d

f. a % b / c \* d

i. (a – (b – c)) – d

**Programming Problems**:

**1.1**:  (no writeup, just do it) Enter, compile, and run the following application:

public class Test

{

public static void main (String[] args)

{

System.out.println ("An Emergency Broadcast");

}

}

**1.2**: (submit the output for each error) Introduce the following errors, one at a time, to the program from the programming project 1.1. Record any error messages that the compiler produces. Fix the previous error each time before you introduce a new one. If no error messages are produced, explain why.

Try to predict what will happen before you make each change.

* + change Test to test
  + change Emergency to emergency
  + remove the first quotation mark in the string
  + remove the last quotation mark in the string
  + change main to man
  + change println to bogus
  + remove the semicolon at the end of the println statement
  + remove the last brace in the program

**1.8**: (submit the code) Write an application that displays your initials in large block letters.

Make each large letter out of the corresponding regular character. For example:

**2.4**: Create a version of the TempConverter application to convert from Fahrenheit to Celsius. Read the Fahrenheit temperature from the user.

**2.6**: Write an application that reads values representing a time duration in hours, minutes, and seconds, and then prints the equivalent total number of seconds. (For example, 1 hour, 28 minutes, and 42 seconds is equivalent to 5322 seconds.)

**2.7**: Create a version of the previous project that reverses the computation. That is, read a value representing a number of seconds, then print the equivalent amount of time as a combination of

hours, minutes, and seconds. (For example, 9999 seconds is equivalent to 2 hours, 46 minutes, and 39 seconds.)