**Exercises:**

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

o Factorial

o anExtremelyLongIdentifierIfYouAskMe

2ndLevel *Identifiers must not start with a number*

o level2

o MAX\_SIZE

o 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?

o q ***One character is too vague for an identifier.***

o totVal ***The word can be spelled out completely for better clarity.***

o 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:

i. multiplying two numbers when you meant to add them ***Logical error.***

j. dividing by zero ***run-time error.***

k. forgetting a semicolon at the end of a programming statement ***compile-time error.***

l. spelling a word wrong in the output ***Logical error.***

m. producing inaccurate results ***Logical error.***

n. 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 in red:

Here we go!12345

Test this if you are not sure.Another.

All done.

//Traces:

//(1) *printed a string without new line first*

*//(2) printed another string with new line right next to the string in (1)*

//(3) On the new line created by (2), *printed without new line twice to join two strings*

//(4) *printed nothing and terminate with a new line,*

*//(5) printed All done. and terminated with new line*

**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.");

//Three missing items:

// (1) a double quote character “ at the *that is the* line,

// (2) a double quote character “ before *the question.* line,

// (3) the + character.

// Change to: 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\"");

*//Output:*

*He thrusts his fists*

*against the post*

*and still insists*

*he sees the "ghost"*

//*Explanation:*

*//(1) Method System.out.println() prints the first string out followed the escape sequence*

*//characters \n (new line) and \t (horizontal tab), followed by the remaining characters (against),*

*//followed by concatenating the second string and printing each character out, including escape sequences*

*// new line character (\n) which creates a new line, some regular characters, another new line (\n) and horizontal*

*//tab (\t) pair of characters which create a new line and printed the subsequent characters 1 tab inward,*

*//followed by regular characters and escape sequence \” to create the double quote, followed by regular*

*//characters and escape sequence \” to unquote and created a new line as the result of System.out.println(); method.*

**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

*(3) (1) (2)*

f. a % b / c \* d

*(1) (2) (3)*

i. (a – (b – c)) – d

*(2) (1) (3)*

**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");

}

}

**//Output**

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

java.lang.NoClassDefFoundError: Test (wrong name: test)

at java.lang.ClassLoader.defineClass1(Native Method)

at java.lang.ClassLoader.defineClassCond(ClassLoader.java:631)

at java.lang.ClassLoader.defineClass(ClassLoader.java:615)

at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:141)

at java.net.URLClassLoader.defineClass(URLClassLoader.java:283)

at java.net.URLClassLoader.access$000(URLClassLoader.java:58)

at java.net.URLClassLoader$1.run(URLClassLoader.java:197)

at java.security.AccessController.doPrivileged(Native Method)

at java.net.URLClassLoader.findClass(URLClassLoader.java:190)

at java.lang.ClassLoader.loadClass(ClassLoader.java:306)

at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:301)

at java.lang.ClassLoader.loadClass(ClassLoader.java:247)

Exception in thread "main"

p. change Emergency to emergency

q. remove the first quotation mark in the string

r. remove the last quotation mark in the string

s. change main to man

t. change println to bogus

u. remove the semicolon at the end of the println statement

v. 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.)