PROGRAMMING POINTERS, LESSON 8

Syntax/correctness issues

8-1 The statement of a control structure can be a single statement, a compound statement marked out with braces {}, or the empty statement that consists of a semicolon (;).

8-2 The condition for a control structure must be placed within parentheses ().

8-3 Be careful when setting up an equality (==) comparison. A very common mistake is the use of the assignment operator (=) when equality was intended. Here is a suggestion, when using comparisons that involve a variable and a value (`x == 2`), reverse the order (`2 == x`) to avoid the subtle error of writing `x = 2`.

Formatting suggestions

8-4 Use consistent indentation when formatting control structures. Indentation implies hierarchy or subordination - which statements belong to which control structure. I suggest three blank spaces per indent.

8-5 When writing expressions with logical and relational operators, add white space around each operator to make the expression more readable. For example:

((*number* <= 10) && (*total* <= 1000)) instead of ((*number*<=10)&&(*total*<=1000))

Software engineering

8-6 Use pseudocode to develop your solution to a problem. Then convert your pseudocode to Java code.

8-7 Programs and subprograms can be broken down into three stages: initialization, processing, and output. When writing a method or an entire program consider this approach:

1. Initialize some variables
2. Solve some processing problem, this usually involves developing an algorithm
3. Return some output to either the screen or to the calling statement of the function.

8-8 The && operator is also a short-circuit operator in Java. If the first operand of an && expression is false, the second condition is not evaluated. Consequently you should write the expression most likely to be false as the first half of an && expression.

```
(expression1 && expression2)
```

If expression1 is false, the && operator will ignore processing expression2.

8-9 The || operator is also an efficient operator. You should put the expression most likely to be true as the first condition of an || expression.

```
(expression1 || expression2)
```

If expression1 is true, the || operator will ignore processing expression2.