PROGRAMMING POINTERS, LESSON 10

Syntax/correctness issues

10-1 Make sure you initialize counters and totals before entering a loop.

10-2 Be careful to avoid infinite loops. For both **`while`** and **`do-while`** loops, make sure that the Boolean expression eventually becomes false.

10-3 Be sure that you are not confusing "less than" with "less than or equal"; or "greater than" with "greater than or equal", when you are coding the Boolean condition for loops.

Formatting suggestions

10-4 Use formatting to indicate subordination of control structures. The statements that belong inside a control structure should be indented about 3 spaces. Good formatting is especially needed with nested control structures to make the code more readable. For example:

```
for (int row = 1; row <= 5; row++)
{
   // if row is odd, print a row of '*', otherwise print a row of '-'
   if (row % 2 == 1)
      c1 = '*';
   else
      c1 = '-';

   for (int col = 1; col <= 10; col++)
      System.out.print(c1);

   System.out.println();
}
```

Software engineering

10-5 Here are some basic guidelines for selecting the appropriate looping tool in Java:

`for` loop – used when the number of iterations can be determined before entering the loop.
`while` loop – used when the loop could potentially occur zero times.
`do-while` loop – used when the loop body should be executed at least once.

10-6 A valuable strategy for developing the Boolean expression for a conditional loop is the idea of negation. This technique consists of two important steps:

1. What will be true when the loop is done? This will be stated as a Boolean expression.
2. Then write the opposite of this Boolean expression.

For example, suppose we wanted to read positive integers from the keyboard until the running total is greater than 50.

What will be true when the loop is done? total > 50
Negate this expression: total <= 50.

We use the negated expression as the boundary condition of our loop.

```
total = 0;
do
{
   System.out.println("Enter an integer ---> ");
   num = keyboard.readInt();
   total += num;
}
while (total <= 50);
```

The expression (`total <= 50`) can be thought of as "keep doing the loop as long as total is less than or equal to 50."

This technique will be developed more completely in a later lesson.