CMPS 161 -- Practice Test #3

1. Write a while loop which will print 2000 asterisks to the screen.

2. Write a loop which will input prices of toys until a price of 0 is entered, and then print out the total cost of all the toys.

3. Fill in the blanks of the following loop which repeats until total becomes more than 75.
   ```java
   int total;
   int total = readInt("Where you like total to start? ");
   ```
   ```java
   { 
   total = total * 2;
   println("Total is up to " + total);
   }
   ```

4. Fill in the blanks of the following loop which continues as long as initial is not a capital X. If a blank is not required, you should leave it blank.
   ```java
   char _________ _________;
   ```
   ```java
   { 
   initial = readLine("What is the new initial? ").charAt(0);
   }
   ```

5. Show exactly what the output of the following program fragment is.
   ```java
   void main()
   { 
   float value = 16;
   while (value > 1 || value < -1)
   { 
   if (value < 0)
   println("Value is "+ value);
   value = value / (-2.0); 
   }
   println("Final value is " + value);
   }
   ```

6. Show exactly what the output of the following program fragment is.
   ```java
   void main()
   { 
   int first = 1;
   int second = 7;
   while (first <= second)
   { 
   println(first * second);
   first = first + 1;
   second = second - 2;
   }
   println(first);
   println(second);
   }
   ```

7. When should you use a do-while loop?
8. When should you use a while loop?

9. When should you use a for loop?

10. For each of the following, write the heading for the for loop described. You need not concern yourself with the body of the loop -- only the heading is required.
   a. Using a control variable called `curPos`, write a for loop heading which will give `curPos` a value of 1 the first time through the loop body, and a value of 1250 the last time through the loop body, and which will increase `curPos` by 1 each time through the loop.

   b. Using a control variable called `curPos`, write a for loop heading which will give `curPos` a value of 5 the first time through the loop body, and a value of 27 the last time through the loop body, and which will increase `curPos` by 2 each time through the loop.

   c. Using a control variable called `curPos`, write a for loop heading which will give `curPos` a value of 1250 the first time through the loop body, and a value of 1 the last time through the loop body, and which will decrease `curPos` by 1 each time through the loop.

   d. Using a control variable called `curPos`, write a for loop heading which will give `curPos` a value of 1 the first time through the loop body, and a value of no more than 500 the last time through the loop body, and which will double `curPos` each time through the loop.

11. What is the output of the following console program fragment? (don't worry about exact spacing)

```java
void main()
{
    int a, b;
    for (a=1; a<5; a++)
    {
        for (b=0; b<a; b++)
            print("*");
        println();
    }
    for (a=5; a>1; a--)
    {
        for (b=0; b<a; b++)
            print("*");
        println();
    }
}
```
12. What is the output of the following console program fragment? (don't worry about exact spacing)

```java
void main()
{
    int k, m;
    k = 1;
    m = 2;
    while ((k<6) && (m<50))
    {
        m = m * 2;
        println(k + " " + m);
        m++;
        k = k + 2;
    }
    println(k + " " + m);
}
```

13. What is the output of the following console program fragment? (don't worry about exact spacing)

```java
void main()
{
    int x, y;
    x = 0;
    y = 0;
    do
    {
        x = x + 2;
        y = x - 2;
        println(x + " " + y);
    }
    while (y <= 5);
}
```

14. Write a method called `distance` to implement the following interface description:

```java
/**
 * Calculate the distance between two Cartesian coordinates. This method uses the standard
distance equation where distance between points (x1, y1) and (x2, y2) is the square root of
x1 squared minus x2 squared plus y1 squared minus y2 squared.
*
* @param x1  The x-coordinate of the first point as a real number.
* @param y1  The y-coordinate of the first point as a real number.
* @param x2  The x-coordinate of the second point as a real number.
* @param y2  The y-coordinate of the second point as a real number.
* @returns  The real number distance between the first and second points.
*/
```
15. Write a method called `box` to implement the following interface description:

```java
/**
 * Display a box on a simple text output medium (such as a Program or
 * ConsoleProgram using asterisks (*’s) to draw just the edges of
 * the box.
 *
 * @param height. The number of asterisks high to make the box.
 * @param width. The number of asterisks wide to make the box.
 * @returns nothing.
 */
```

16. Write a method called `askNumberQuestion` to implement the following interface description:

```java
/**
 * Ask a question whose answer is a real number with a maximum and
 * minimum value. This method asks a question and gets a response
 * from the user. If the response is within the maximum and minimum
 * values (inclusive) it will be accepted, and the value input will be
 * returned. Otherwise, an error message is displayed and the question
 * re-asked until a valid response is obtained.
 *
 * @param question. A string containing the yes or no question to be
 * asked.
 * @param min. The minimum response to this question, as a real
 * number.
 * @param max. The maximum response to this question, as a real
 * number.
 * @returns Either a capital Y character or a capital N
 * character, depending on if the answer was yes or no.
 */
```
17. Write a main method which will ask the user for x and y real-number values for 2 points, where the x values must be between 2.5 and 1000, and the y values must be between -500 and +500. It will then calculate the distance between these two points and display the result. Use the askNumberQuestion and distance methods described above to implement this main method.

18. Write a main method which will ask the user for a number of rows and a number of columns, where the number of rows or columns must be at least 1 and no more than 50. It will then convert the dimensions to integers, and display a box with these dimensions. Use the askNumberQuestion and box methods described above to implement this main method.