CMPS 161 -- Practice Test #3 Key

1. `int asteriskCount = 0;`  
   `while (asteriskCount < 2000)`  
   `{`  
   `    print('*');`  
   `    asteriskCount = asteriskCount + 1; // or asteriskCount++`  
   `}`

2. `float toyPrice=1000.00F;`  
   `float totalCost = 0.0F;`  
   `while (toyPrice != 0)`  
   `{`  
   `    toyPrice = (float)readDouble("Toy price: ");`  
   `    totalCost = totalCost + toyPrice;`  
   `}`

3. `int total;`  
   `int total = readInt("Where you like total to start? ");`  
   `while (total <= 75)`  
   `{`  
   `    total = total * 2;`  
   `    println("Total is up to " + total);`  
   `}`

4. `char initial = 'Z';`  
   `while (initial != 'X')`  
   `{`  
   `    initial = readLine("What is the new initial? ").charAt(0);`  
   `}`

5.  
   Value is -8.000000  
   Value is -2.000000  
   Final value is 1.000000

6.  
   7  
   10  
   9  
   4  
   1

7. When should you use a do-while loop?  
   
   *When you don't know how many times the loop will execute, and you want it to execute the body at least once.*

8. When should you use a while loop?  
   
   *When you don't know how many times the loop will execute, and it might not execute the body at all.*
9. When should you use a for loop?

*When you know beforehand (or the program can calculate prior to the loop execution) the number of times that the loop should execute.*

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.

```plaintext
for (curPos=1; curPos <=1250; curPos++)
```

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.

```plaintext
for (curPos=5; curPos <=27; curPos = curPos + 2)
```

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.

```plaintext
for (curPos=1250; curPos >= 1; curPos--)
```

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.

```plaintext
for (curPos=1; curPos <=500; curPos = curPos * 2)
```

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

```plaintext
void main()
{
 int a, b;
 for (a=1; a<5; a++)
  {
   for (b=0; b<a; b++)
     print("*" as string);
   println();
  }
 for (a=5; a>1; a--)
  {
   for (b=0; b<a; b++)
     print("*" as string);
   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++; // Changed from 'm++' to prevent infinite loop
        k = k + 2;
    }
    println(k + " " + m);
}
```

```
1 4
3 10
5 22
7 23
```

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);
}
```

```
2 0
4 2
6 4
8 6
```

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.
 */

double distance(double x1, double y1, double x2, double y2)
{
    return Math.sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));
}
```
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 the 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.
 */
void box(int height, int width)
{
    int curRow, curCol;
    for (curCol = 0; curCol < width; curCol++)
        print('*');
    println();
    for (curRow = 0; curRow < height - 2; curRow++)
    {
        print('*');
        for (curCol = 0; curCol < width - 2; curCol++)
            print(' ');
        println('*');
    }
    for (curCol = 0; curCol < width; curCol++)
        print('*');
    println();
}
```

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.
 */
double askNumberQuestion(String question, double min, double max)
{
    double answer;
    boolean badAnswer;
    do
    {
        answer = readDouble(question);
        badAnswer = answer < min || answer > max;
        if (badAnswer)
            println("The answer must be from " + min + " to " + max + ").;
    }
    while (badAnswer);
    return answer;
```
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.

```java
void main()
{
    double startX = askNumberQuestion("What is the x-coordinate of the starting point? ", 2.5, 1000);
    double startY = askNumberQuestion("What is the y-coordinate of the starting point? ", -500, 500);
    double endX = askNumberQuestion("What is the x-coordinate of the ending point? ", 2.5, 1000);
    double endY = askNumberQuestion("What is the y-coordinate of the ending point? ", -500, 500);
    double curDistance = distance(startX, startY, endX, endY);
    println("The distance between the points is "+ curDistance);
}
```

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.

```java
void main()
{
    int numRows = (int)askNumberQuestion("How many rows in the box? ",1, 50);
    int numCols = (int)askNumberQuestion("How many columns in the box? ",1, 50);
    box(numRows, numCols);
}
```