

## WES-CS GROUP MEETING #9

### Exercise 1: Code Tracing

The code segment shown below was supposed to set the *characters* array to hold the characters

```
a * b * c * d * e * f * g * h * i * j *
```

However, the code does not work correctly.

```
char [] characters = new char [20];
for (int x = 0; x < characters.length; x++) {
    characters[x] = '*';
}

char a = 'a';
for (int x = 0; x < characters.length; x++) {
    while (a != 'k') {
        if (x%2==0) characters[x] = a;
        a = (char)(a + 1);
    }
}
```

First, trace the code to find out what the final values in the *characters* array actually are.

Then, fix the code so that it works as intended.

## Exercise 2: Nested Loops

The six pictures shown below were printed by a simple Java program.

```
*
**
***
****
*****
*****
*****

*
***
*****
*****
*****
*****
*****

*****
*
*
*
*
*
*
*****

*
**
*
*
*
*
*
*****

*****
**
*
*
*
*
*
*****

*****
**
*
*
*
*
*
*****
```

The first picture was printed by the program shown below.

```
public class Loops {
    public static void printDesign1( int n ) {
        // print n rows of increasing length
        for (int row=0; row<n; row++) {
            for (int col=0; col<row+1; col++) {
                System.out.print('*');
            }
            System.out.println();
        }
        System.out.println();
    }

    public static void main(String[] args) {
        int n = 7;
        printDesign1( n );
    }
}
```

Trace the code to understand how it works. Then choose one or more of the other pictures, and add new methods that print the pictures you've chosen. (Make sure that your methods work for any value of  $n$  greater than 0.)

Now design your own picture (some pattern of stars in an  $n$ -by- $n$  box) and ask your neighbor to code it.

*Space for answers to Exercise 2*

### Exercise 3: Arrays

Divide into two teams to play a game like Jeopardy with 100 and 200-point questions (three questions in each category), plus one *final Jeopardy* question. The questions involve the six code fragments given below. Assume that variable `arr` is a non-empty, initialized array of `int` values.

For each question, the team that grabs the flashing light thingy first gets to try to answer the question. If the answer is right, the team gets the points; otherwise, the team loses that many points.

Fragment 1:	<pre>for (int k=0; k &lt; arr.length; k++) {     if ( _____ ) arr[k] = 0; }</pre>
Fragment 2:	<pre>for (int k=1; k &lt; arr.length; k++) {     if ( _____ ) return false; } return true;</pre>
Fragment 3:	<pre>for (int k=0; k &lt; arr.length; k++) {     if ( _____ ) return false; } return true;</pre>
Fragment 4:	<pre>for (int j=0; j &lt; arr.length; j++) {     for (int k=j+1; k &lt; arr.length; k++) {         if ( _____ ) return false;     } } return true;</pre>
Fragment 5:	<pre>for (int j=0; j &lt; arr.length-1; j++) {     if ( _____ ) swap(arr, j, j+1); } }</pre>
Fragment 6:	<pre>for (int j=0; j &lt; _____; j++) {     if ( _____ ) return false; } return true;</pre>

## Exercise 4: Logical Thinking

Today's logical-thinking exercise is an old chess puzzle. The board (shown below) is a 3x3 part of the chessboard. The goal is to swap the red and blue knights, using a sequence of legal moves (of course, two knights may not occupy the same square at any time).

See who can find the *shortest* sequence of moves. Can you find a good argument that this is the minimum number of moves required to solve the puzzle?

**B**: blue knight

**R**: red knight

<b>B</b>		<b>B</b>
<b>R</b>		<b>R</b>

### 100 Point Array Questions

Q: How could you fill in the blank in Code Fragment 1, so that when the code finishes, every position that originally had an even value ends up with the value zero?

A: `arr[k]%2 == 0, or (arr[k]/2 * 2) == arr[k]`

Q: How could you fill in the blank in Code Fragment 2, so that the code returns `true` iff the value in `arr[0]` doesn't occur in any other place in the array?

A: `arr[0] == arr[k]`

Q: How could you fill in the blank in Code Fragment 2, so that the code returns `true` iff the values in the array are in sorted order, from low to high?

A: `arr[k-1] > arr[k]`

### 200 Point Array Questions

Q: How could you fill in the blank in Code Fragment 3, so that the code returns `true` iff the value in `arr[0]` doesn't occur in any other place in the array?

A: `(k != 0) && (arr[0] == arr[k])`

Q: How could you fill in the blank in Code Fragment 4, so that the code returns `true` iff the array contains no duplicate values (i.e., no value is stored in more than one place in the array).

A: `arr[j] == arr[k]`

Q: Assume that the call `swap(arr, x, y)` swaps the value in `arr[x]` with the value in `arr[y]`. How could you fill in the blank in Code Fragment 5 so that when the code finishes executing, the largest value is at the end of the array?

A: `arr[j] > arr[j+1]`

### Final Jeopardy Question

Q. How could you fill in the two blanks in Code Fragment 6, so that the code returns `true` iff the sequence of values in the array from left to right is the same as the sequence of values from right to left?

A: First blank: `arr.length / 2`  
Second blank: `(arr[j] != arr[arr.length-(j+1)])`