Quiz 18: Arrays

Name: _____

CSCI 110 Section 1

Friday, October 7, 2016

1) Write a function to print out the contents of an array line-by-line. You can choose the type of array. [warmup]

Return type? void - don't need to return anything
Parameters? The array we want to print

```
void printContents(int[] arr) {
	for (int i = 0; i < arr.length; i++) {
		System.out.println(arr[i]);
	}
}
```

2) Write a function that returns the sum of all of the elements of an integer array. [30 points]

Return type? int, since we want to return the sum of a bunch of integers
Parameters? the array we want to sum

```
int sumArray(int[] arr) {
	int sum = 0;
	for (int i = 0; i < arr.length; i++) {
		sum += arr[i];
	}
	return sum;
}
```

3) Write a function that fills an integer array with the numbers 1 through 10, then returns it. [30 points]

Return type? int array
Parameters? Nothing, we're always going to return the same array

```
int[] getOneThroughTenArray() {
	int[] result = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
	return result;
}
```

4) Write a function that fills an integer array starting at 1 and ending at a user-specified upper bound (inclusive) [40 points]

Return type? An int array
Parameters? The upper bound

```
int[] makeArray(int bound) {
	int[] result = new int[bound];
	for (int i = 0; i < bound; i++) {
		result[i] = i+1;
	}
	return result;
}
```

5) Write a function that fills an integer array starting at a user-specified lower bound and ending at a user-specified upper bound (inclusive if it happens to end on the upper bound), skipping every other number. [extra credit, 25 points]

Return type? an int array
Parameters? The lower and upper bounds

For this one, we use two variables as we're going through:

- An iterator i, which starts at 0 and increments by 1
- A value variable that's named "curr" below, which starts at the user-provided lower bound and increments by 2

```
int[] makeEveryOtherArray(int lower, int upper) {
	int[] result = new int[(upper - lower) / 2 + 1];
	int curr = lower;
	for (int i = 0; i < result.length; i++) {
		result[i] = curr;
		curr += 2;
	}
	return result;
}
```