Name _____

CISC 3115 – Modern Programming Techniques
Spring 2025
Quiz 0

1. a. (75%) Write a boolean-valued method, `areDescending`, that accepts three integer parameters and returns `true` if they are in descending order. For example, `5, 3, 2` and `10, 8, 8` both return true, while `7, 6, 8` returns `false` (note that adjacent equal parameters do not violate the descending condition)

```
public static boolean areDescending(int n1, int n2, int n3) {
        return n1 >= n2 && n2 >= n3;
}
```

For those of you who don't 'talk boolean':

```
public static boolean areDescending(int n1, int n2, int n3) {
        if (n1 >= n2 && n2 >= n3)
                return true;
        else
                return false;
}
```

b. (25%) Assuming the availability of a boolean-valued method, `allAreEqual`, that accepts three integer parameters and returns `true` if all three parameters are equal, write a boolean-valued method `twoAreEqual` that returns `true` if *exactly* two of the parameters are equal (i.e., it returns `false` if all three are equal). For full credit you *must not* use a conditional statement or operator, and you *must* use `allAreEqual`.

```
public static boolean twoAreEqual(int n1, int n2, int n3) {
        return !allAreEqual(n1, n2, n3) &&
                (n1 == n2 || n1 == n3 || n2 == n3);
}
```

2. a. (75%) Assuming a reference variable, `sc`, has been declared and initialized to a `Scanner` object, read in a sequence of integers consisting of a header value followed by that many integers, e.g.

```
5  3 6 2 4 7
```

and print out the sequence followed by its average. Do not use an array or any other data structure.

```
int howMany = scanner.nextInt();
System.out.print("The average of the " + howMany + " integers ");
int total = 0;
for (int i = 0; i < howMany; i++) {
      int number = scanner.nextInt();
      total += number;
      System.out.print(number + " ");
}
double average = total / (double)howMany;
System.out.println("is " + average);
```

b. (25%) Suppose you were asked to print out the average followed by the sequence of numbers (i.e., the opposite order from *part a*), how would that change your code (if at all)? Explain your answer.

Since the elements can't be printed out until the average has been calculated (i.e., all have been read in), you would need an array.

3. a. (75%) Write a method that accepts a filename and a value and prints out the position of the last occurrence of the value in the file (the first position is 1). If the value is not in the file, print out an appropriate message. You should not use an array or any other data structure.

```
public static int findLast(String filename, int value) throws Exception {
      Scanner scanner = new Scanner(new File(filename));

      int count = 0, result = 0;

      while (scanner.hasNextInt()) {
           int number = scanner.nextInt();
           count++;
           if (number == value) result = count;
      }
      if (count == 0)
           System.out.println("Found " + value + " at " + result);
      else
           System.out.println(value + "not found"
  }
```

b. (25%) How would printing out the first occurrence (rather than the last) change your logic?

As soon as the desired item is encountered, one could leave the method and return that position

4. a. (75%) The file `numbers.text` contains a header value (an integer), followed by that many double values. Create an array of the appropriate size (i.e., the size of the sequence of doubles), populate the array with the doubles, and print the middle and largest elements of the array. The largest element should be calculated using a method.

```
int capacity = scanner.nextInt();

double [] arr = new double[capacity];

for (int i = 0; i < arr.length; i++)
    arr[i] = scanner.nextDouble();

System.out.println("The middle element of the array is " +
    arr[(arr.length-1)/2);
System.out.println("The largest element is " + arr[maxPos(arr)]);



public static int maxPos(double [] arr) {
    int result = 0;
    for (int i = 1; i < arr.length; i++)
       if (arr[i] > arr[result]) result = i;
           return result;
}
```

b. (25%) Suppose the file did not contain a header, but you were to read to eof. How does that change your logic?

Would have to introduce a size variable and use partially populated array logic

Optional: (10%) What happens to your max method if the array is empty (i.e, has no values)?

Would either return an invalid value, or (as in the above code) result in the caller having an ArrayIndexOutOfBoundsException