Name (print): __________________________________________

• Instructions:
  – Keep your eyes on your own paper and do your best to prevent anyone else from seeing your work.
  – Do NOT communicate with anyone other than the professor/proctor for ANY reason in ANY language in ANY manner.
  – This exam is closed notes, closed books, and no calculator.
  – Turn all mobile devices off and put them away now. You cannot have them on your desk.
  – Write neatly and clearly indicate your answers. What I cannot read, I will assume to be incorrect.
  – Stop writing when told to do so at the end of the exam. I will take 5 points off your exam if I have to tell you multiple times.
  – Academic misconduct will not be tolerated. Suspected academic misconduct will be immediately referred to the Emory Honor Council. Penalties for misconduct will be a zero on this exam, an F grade in the course, and/or other disciplinary action that may be applied by the Emory Honor Council.

• Time: This exam has 7 questions on 9 pages including the title page. Please check to make sure all pages are included. You will have 75 minutes to complete this exam.

I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Emory community. I have also read and understand the requirements and policies outlined above.

Signature: __________________________________________

<table>
<thead>
<tr>
<th>Question</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>16</td>
<td>8</td>
<td>11</td>
<td>12</td>
<td>75</td>
</tr>
<tr>
<td>Score:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. Parameter Passing:

   (a) (3 points) Explain the difference between “pass by reference” and “pass by value” for method calls.

   Solution: pass by value: parameters are copied and changes made within a method only affect the copied values
   pass by ref: parameter variables are really addresses (references) to a location in memory, so changes made within a method affect the values outside of the method as well.

   (b) Consider the following code:

   ```java
   public class Parameters {
       public static int p(int a, int b) {
           a = a + 4;
           b = b - 2;
           return a*b;
       }
       public static void main(String[] args) {
           int x = 1;
           int y = 3;
           int z = p(x,y);
           System.out.println("x: " + x);
           System.out.println("y: " + y);
           System.out.println("z: " + z);
       }
   }
   
   (a) (3 points) What would the code print if the parameters were passed by value?

   Solution:
   x: 1
   y: 3
   z: 5

   (b) (3 points) What would the code print if the parameters were passed by reference?

   Solution:
   x: 5
   y: 1
   z: 5
2. Consider the function Mystery which is given below. This function has limitations on its input: number must be greater than 0 and contain no 0s as digits. value must be greater than 0 but less than 10.

```
public int Mystery(int number, int value) {
    int count = 0;
    int dig = number % 10;
    while (dig > 0) {
        if (dig == value) {
            count++;
        }
        number = number/10;
        dig = number % 10;
    }
    return count;
}
```

Answers the following questions.
(a) (1 point) What type is the return type of Mystery?

Solution: int

(b) (1 point) How many local variables does Mystery have and what is/are their names?

Solution: 2; count, dig

(c) (1 point) How many parameter variables does Mystery have and what is/are their names?

Solution: 2; number and value

(d) (2 points) What is the return value for the call Mystery(376646, 6)?

Solution: 3

(e) (2 points) What is the return value for the call Mystery(376646, 5)?

Solution: 0

(f) (2 points) Describe in words what the method Mystery does.

Solution: counts the number of digits in number that are equal to value
3. (10 points) Write the output that the following code will generate. You may assume the code compiles as written.

```java
public class ExamQuestion {
    public static int a = 10;

    public static int method1(double d) {
        System.out.println("Location3");
        if (a < 10 && d < 5) {
            return a + 4;
        } else {
            a = a + 2;
        }
        System.out.println("a in method1: " + a);
        return a;
    }

    public static int method2(int a) {
        System.out.println("Location2");
        a = a + 10;
        System.out.println("a in method2: " + ExamQuestion.a);
        return a;
        System.out.println("Location3");
    }

    public static void method3() {
        for(int i = 0; i < 6; i++) {
            int a = i + 6;
            if(i == 4) {
                System.out.println(a);
            }
        }
        System.out.println("a in method3: " + a);
    }

    public static void main(String[] args) {
        System.out.println("a1: " + a);
        int a = method1(3.14);
        System.out.println("a2: " + a);
        int b = method2(4);
        System.out.println("b: " + b);
        method3();
    }
}
```
Solution:

a1: 10
Location3
a in method1: 12
a2: 12
Location2
a in method2: 12
b: 14
10
a in method3: 12
4. Draw the array that would result after the following code is executed.

(a) (4 points) int[] data = new int[8];
data[0] = 3;
data[7] = -18;
data[4] = 5;
data[1] = data[0];

int x = data[4];
data[4] = 6;
data[x] = data[0] * data[1];

Solution: [3,3,0,0,6,9,0,-18] Some students left out 0s (initial values when array is created). -1 for leaving out all 0s.

(b) (6 points) int[] list = {2,18,6,−4,5,1};
for (int i = 0; i < list.length; i++) {
    list[i] = list[i] + (list[i] / list[0]);
}

Solution: [3,24,8,−5,6,1]

(c) (6 points) Consider the function below:
public static void mystery(int[] a) {
    for(int i = 0; i < a.length - 1; i++) {
        if(a[i] < a[i+1]) {
            a[i] = a[i+1];
        }
    }
}

Draw the array a2 after the code below executes.
int[] a2 = {2, 4, 6, 3, 7, 9};
mystery(a2);

Solution: [4,6,6,7,9,9]
5. Professor Summet wrote the following method to reverse the data in an array.

```java
public static void reverse(int[] a) {
    for (int i = 0; i < a.length; i++) {
        int temp = a[i];
        int swapIndex = a.length - 1 - i;
        a[i] = a[swapIndex];
        a[swapIndex] = temp;
    }
}
```

However, she wasn’t careful and her code contains a logical error. That is, it compiles and runs, but doesn’t do what she intended. In fact, she can see no change in her array at all after the method executes!

(a) (4 points) Explain the logical error in this code. You can draw pictures if it helps your explanation.

**Solution:** Prof. Summet the first half of the array, but then proceeds through the last half of the array and “reswaps” the elements back to their original state.

(b) (4 points) Rewrite the method to fix Professor Summet’s logical error.

**Solution:** The easiest thing to do is to modify the code so that the loop executes half as many times (and thus, only swaps the “front” values of the array with the “back” values of the array) rather than re-swapping the values. Using this idea, the loop would be:

```java
for(int i = 0; i < a.length/2; i++)
```

You could also write a solution using another array, but you would need to be sure to add the statement `a = <other array variable>;` to ensure your code modified `a`. 

6. (11 points) Write a function \(\text{no29}\) which takes an array of single digit integers as a parameter. The function should return a boolean value. The function returns true if the array contains no 2s or it contains no 9s. Examples of function calls and return values:

\[
\begin{align*}
\text{no29}(&\{1,2,3\}) \text{ returns true} \\
\text{no29}(&\{1,2,8,9\}) \text{ returns false} \\
\text{no29}(&\{7,8,9\}) \text{ returns true}
\end{align*}
\]

Solution:

```java
public static boolean no29(int[] a) {
    boolean no2 = true;
    boolean no9 = true;

    for (int i = 0; i < a.length; i++) {
        if (a[i] == 2) {
            no2 = false;
        }
        if (a[i] == 9) {
            no9 = false;
        }
    }

    return no2 || no9;
}
```

+3: function header (return value, name, parameters)
+2: uses loops correctly to examine all elements
+2: detects no 2s
+2: detects no 9s
+1: returns boolean
+1: returns correct value

Another way of solving the problem: count all the twos and count all the 9s in the array. Return false if the 2’s count is \(\geq 0\) AND the 9’s count is \(\geq 0\). For example:

\[
\text{if (two_count > 0 \&\& nine_count > 0) return false;}
\]

Otherwise, return true.
7. (12 points) Write a function `indexOutOfOrder`. The function should take an array of integers as an input parameter. That array is supposed to be in increasing order. The function should return the index of the first number in the array that is out of order. If none of the numbers are out of order, it should return -1. For example, if the array below was the input parameter, a call to `indexOutOfOrder` should return 3 since 7 (which is in position 3) is out of order.

| 3 | 4 | 9 | 7 | 12 | 11 | 18 | 10 |

**Solution:**

```java
public static int indexOutOfOrder(int[] x) {
    for (int i = 1; i < x.length; i++) {
        if (x[i] < x[i-1]) {
            return i;
        }
    }
    return -1;
}
```

Note: this function could be written either by looking back and comparing the current element to the previous one (the solution above) or by looking ahead and comparing the current element to the next element. The difference would be in where your loop starts and the comparison condition.

**Scoring:**
+3: function header (return type, name, parameters)
+2: uses loop correctly (including terminating loop if out of order data is found)
+1: no out of bounds error (correctly accounts for first/last position in loop/array)
+2: compares current position in loop to next (or previous position)
+2: returns correct out of order element
+2: returns -1 if all in order