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 10 pages including the title page. Please check to make sure all pages are included. You will have 50 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>4</td>
<td>4</td>
<td>10</td>
<td>6</td>
<td>6</td>
<td>10</td>
<td>10</td>
<td>50</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. Consider the function `mysteryMethod` which is given below.

```java
public static boolean mysteryMethod(int num1, int num2, int num3) {
    if (num2 < num1) { return false; }
    if (num3 < num1) { return false; }
    if (num3 < num2) { return false; }
    return true;
}
```

(a) (1 point) What is the return type of `mysteryMethod`?

**Solution:** boolean

(b) (1 point) What is the return value for the call `mysteryMethod(256, 5000, 4990)`?

**Solution:** false

(c) (2 points) Describe in your own words what `mysteryMethod` does.

**Solution:** This mysteryMethod takes in three integers (num1, num2, num3, in that order). If num1 ≤ num2 ≤ num3, then the method returns true, otherwise it returns false. In other words, checks to see if the numbers were given in increasing order.

2. Consider the function `mysteryMethod2` which is given below.

```java
public static double mysteryMethod2(int num1, int num2, int num3) {
    double out = num1;
    if (num2 < num1 && num2 < num3) { out = num2; }
    if (num3 < num1 && num3 < num2) { out = num3; }
    return out;
}
```

(a) (1 point) How many local variables does `mysteryMethod2` have, and what is/are their names?

**Solution:** One local variable: out

(b) (1 point) How many parameter variables does `mysteryMethod2` have, and what is/are their names?

**Solution:** Three parameter variables: num1, num2, and num3
(c) (1 point) What is the return value for the call \texttt{mysteryMethod2(256, 5000, 4990)}?

\begin{center}
\textbf{Solution:} 256.0
\end{center}

(d) (1 point) Describe in your own words what \texttt{mysteryMethod2} does.

\begin{center}
\textbf{Solution:} \texttt{mysteryMethod2} returns the minimum of the three numbers passed to it.
\end{center}
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; // a is available anywhere in the class
    public static void method1() {
        System.out.println("Location1");
        for (int i=0; i < 8; i++) {
            if (i == 6) {
                System.out.println(a);
            }
            a = i * 10;
        }
        System.out.println("a in method1: " + a);
    }
    public static int method1(int b) {
        System.out.println("Location2");
        b = b + 3;
        return b;
        System.out.println("Location3");
    }
    public static int method2(double c) {
        System.out.println("Location4");
        if (a < 10 && c < 10.0) {
            return a + 5;
        } else {
            a = a + 2;
        }
        System.out.println("a in method2: " + a);
        return a;
    }
    public static void main(String[] args) {
        int b = 0;
        a = method1(1);
        System.out.println("a1: " + a);
        System.out.println("b: " + b);
        double c = method2(9.9);
        System.out.println("a2: " + a);
        System.out.println("c: " + c);
        method1();
        System.out.println("a3: " + a);
    }
}
```

4
Solution:

Location2
a1: 4
b: 0
Location4
a2: 4
c: 9.0
Location1
50
a in method1: 70
a3: 70

Rubric: 1 point for each correct printed line.
-1 point for printing a line (or lines) that should not be reached: (e.g. printing Location3)
-0.5 for c: 9 or c: 12.0 on the line that should be c: 9.0
4. In each case below, draw the array that would result after the code is executed.

(a) (3 points) int[] data = new int[10];
    data[0] = 5;
    data[4] = 25;
    data[8] = data[2];
    int i = data[0];
    data[0] = data[4];
    data[i] = data[2] + data[4];

    Solution: {25, 0, -15, 0, 25, 10, 0, -15, 0}

(b) (3 points) int[] list = {3, 6, 8, 20, 4};
    for (int i = 2; i < list.length; i++) {
        list[i] = list[i] + (list[i-1] / list[i-2]);
    }

    Solution: {3, 6, 10, 21, 6}
5. Consider the following class `Rectangle`

```java
public class Rectangle {
    double length = 0;
    double width = 0;

    Rectangle(double s) {
        length = s;
        width = s;
    }

    Rectangle(double l, double w) {
        length = l;
        width = w;
    }

    double getArea() { return length * width; }
}
```

(a) (1 point) How many data fields does `Rectangle` have and what is/are their names?

**Solution:** 2 data fields: `length` and `width`

(b) (1 point) Consider the following line of code: `Rectangle rectangle1;`
Fill in the blanks using the terms “type”, “class”, or “variable”:
This line of code declares a __________ rectangle1 to be of __________ Rectangle.

**Solution:** This line of code declares a variable `rectangle1` to be of type `Rectangle`.

(c) (1 point) Consider the following line of code:
`Rectangle rectangle1 = new Rectangle(3, 4.5);`
State the values of the data field(s) of `rectangle1`

**Solution:**
length: 3.0
width: 4.5

(d) (2 points) Consider the following line of code:
`Rectangle rectangle1 = new Rectangle();`
Explain why this line of code would give a syntax error (fail to compile).
Solution: Since the class definition for Rectangle includes one or more constructors, the JVM provides no default constructor. Therefore, we only have available the constructors written in the class definition; since this does not include a no-argument constructor, this line of code will cause a syntax error. (Note that if no constructors had been included in the definition, then the JVM would have provided a default constructor, which is a no-argument constructor.)

(e) (1 point) Write a line of code that uses the single-argument constructor to create a rectangle object named square with both sides of length 2.

Solution: Rectangle rectangle1 = new Rectangle(2);
or Rectangle rectangle1 = new Rectangle(2.0);
6. (10 points) Write a method **indexOfMaximum**. The method should take an array of doubles as an input parameter. The function should return the position of the greatest value in the array.

Sample output:

```
indexOfMaximum({1.0, -2.5, 11.111, 0.0}) returns 2
```

Solution:

```java
public static int indexOfMaximum(double[] list) {
    int position = 0;
    for (int i=1; i < list.length; i++) {
        if (list[i] > list[position]) {
            position = i;
        }
    }
    return position;
}
```
7. (10 points) Write a method `absoluteValues`. The method should take an array of doubles as an input parameter. The function should return an array containing the absolute values of the numbers in the input array. (Note, your method should leave the input array unchanged.)

Sample output:

```
absoluteValues({1.0, -2.5, 11.111, 0.0}) returns {1.0, 2.5, 11.111, 0.0}
```

**Solution:**

```
public static double[] absoluteValues(double[] list) {
    double[] abslist = new double[list.length];
    for (int i=0; i < list.length; i++) {
        if (list[i] < 0)
            abslist[i] = -1*list[i];
        else
            abslist[i] = list[i];
    }
    return abslist;
}
```