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 9 questions on 12 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>8</th>
<th>9</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points:</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>17</td>
<td>7</td>
<td>10</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>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. (1 point) Circle the appropriate answers in the problem below:

I HAVE / HAVE NOT checked my final exam schedule as Prof. Summet requested in class and in her reminder email.

If I have checked, I DO / DO NOT have a conflict with the CS 170 final exam period.

If I have a conflict I HAVE / HAVE NOT emailed Prof. Summet with my documentation.

If I have not checked my final exam schedule, I promise to do so immediately following this exam.

2. Parameter Passing. Consider the following code:

```java
public class Parameters {
    public static int p(double a, double b) {
        a = a + 1.0;
        b = b - 2.0;
        return (int)(a + b);
    }
    public static void main(String[] args) {
        double a = 2.3;
        double b = 6.9;
        int c = p(a,b);
        System.out.println("a: " + a);
        System.out.println("b: " + b);
        System.out.println("c: " + c);
    }
}
```

(a) (3 points) What would the code print if the parameters were passed by value?

**Solution:** a: 2.3 b: 6.9 c: 8 (most common error was not casting from double to int for return value)

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

**Solution:** a: 3.3 b: 4.9 c: 8
3. Consider the two functions below which are part of a Java program.

```java
public static void colors(String red, String green, String blue) {
    System.out.println(red + " is better than " + blue);
    System.out.println("But " + green + " is my favorite color");
}

public static void main(String[] args) {
    String red = "pink";
    String green = "yellow";
    String blue = "purple";
    
    colors(green, blue, red);
}
```

Answers the following questions.

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

**Solution:** void / no return type

(b) (1 point) How many local variables does the `main` method have and what is/are their names?

**Solution:** 3; red, green, blue

(c) (1 point) How many parameter variables does the function `colors` have and what is/are their names?

**Solution:** 3; red, green, blue

(d) (2 points) What does the program print out when it is run?

**Solution:** yellow is better than pink
but purple is my favorite color
4. (7 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 = -6;
    public static boolean b = true;

    public static int method1(boolean b) {
        System.out.print("a method1 ");
        if (b) {
            a += 3;
        } else {
            int a = 2;
            System.out.println(a);
            return a + 4;
        }
        System.out.println(a);
        return a;
    }

    public static int method2(int a) {
        a = a + 10;
        ExamQuestion.a -= 1;
        System.out.println("a1 in method2: " + ExamQuestion.a);
        System.out.println("a2 in method2: " + a);
        return a-1;
    }

    public static void method3() {
        a = 2;
        a = ExamQuestion.a + a;
        System.out.println("a2 in method3: " + a);
    }

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

4
Solution:

main a1: -6
a method1 2
main a2: 6
a1 in method2: -7
a2 in method2: 14
main b: 13
a2 in method3: 4
5. (10 points) For each entry below, state whether the code is correct as written or has an error. Show where the error is (you can circle the incorrect code) and explain why it is an error.

<table>
<thead>
<tr>
<th>Code</th>
<th>Error?</th>
<th>Why?</th>
</tr>
</thead>
<tbody>
<tr>
<td>String a = &quot;100&quot;; for (int i = 0; i &lt; a.length(); i++) System.out.println(a[i]);</td>
<td>Yes</td>
<td>Can’t access Strings like arrays. If you want to print a specific character, you need to use a.charAt(i).</td>
</tr>
<tr>
<td>public static int foo(int[] a) { return a[a.length]; }</td>
<td>Yes</td>
<td>Will generate a runtime error: ArrayIndexOutOfBoundsException because a.length is too large.</td>
</tr>
<tr>
<td>if (true &amp;&amp; false) { int a = 3; } a++;</td>
<td>Yes</td>
<td>a is declared inside if stmt. Thus, a++ will be using an out of scope variable</td>
</tr>
<tr>
<td>int[] a = new int[]; a[0] = 10;</td>
<td>Yes</td>
<td>Array declaration is missing the size (number of elements) of the array</td>
</tr>
<tr>
<td>public static int foo(int a) { System.out.println(a); }</td>
<td>Yes</td>
<td>Function needs a return value or needs to be declared as void</td>
</tr>
</tbody>
</table>
6. Draw the array that would result after the following code is executed.

(a) (5 points) boolean[] data = new boolean[5];
    data[0] = true;
    data[2] = !data[1];
    data[4] = true;
    data[1] = false;

Solution: [true, false, true, false, true]

(b) (6 points) int[] li = {3, 19, 6, 0, 5, -1};
    for (int i = li.length-1; i >= 0; i-=2) {
        li[i] = li[i] - (li[i] % li[0]);
    }

Solution: i = 5, 3, 1
[3, 18, 6, 0, 5, 0]

(c) (6 points) int[] a = {-3, 0, -1, 4, 8, 2};
    int m = a[0];
    for(int i = 1; i < a.length; i++) {
        if(a[i] > m) {
            int t = a[i];
            a[i] = m;
            m = t;
        }
    }

Solution: [-3, -3, -1, 0, 4, 2]
7. (7 points) Write a function `fives` which takes an array of integers as a parameter. The function should modify the array. Specifically, the function should make all the numbers in the array a multiple of 5. If the data in the array at a given location is already a multiple of 5, it should not be changed. Examples of function calls and array values after function call:

- `fives({10,2,3})` modifies the array to `{10, 10, 15}
- `fives({1,2,8,9})` modifies the array to `{5, 10, 40, 45}
- `fives({70,5,9})` modifies the array to `{70, 5, 45}

Solution:

```java
public static void tens(int[] a) {
    for (int i = 0; i < a.length; i++) {
        if(a[i] % 5 != 0) {
            a[i] = a[i] * 5;
        }
    }
}
```

+2: function header (name, parameter, return type)
+2: uses loop correctly to examine all elements
+2: changes numbers to multiple of 5
+1: doesn’t change numbers if already multiple of 5

Most common error: returning array (function says to just modify array)
8. (10 points) Write a function `oddEven`. The function takes an array of integers as a parameter. It should return an array with the same data but rearranged so that all the odd numbers come before the even numbers. Other than this restriction, the numbers can be in any order. You can modify and return the given array or make a new array. Examples:

```
oddEven({1,0,1,1,0,1}) returns {1,1,1,0,0,0}
oddEven({3,2,2,3}) returns {3,3,2,2}
oddEven({2,2,2}) returns {2,2,2}
```

**Solution:** Solution using extra array:

```java
public int[] evenOdd(int[] nums) {
    int oddindex = 0;
    int evenindex = nums.length-1;

    int[] a = new int[nums.length];

    for(int i = 0; i < nums.length; i++) {
        int number = nums[i];
        if(number % 2 == 0) {
            a[evenindex] = number;
            evenindex--;
        } else {
            a[oddindex] = number;
            oddindex++;
        }
    }
    return a;
}
```

Solution using same array (modifying array)

```java
public static void oddEven(int[] nums) {
    int evenindex = 0; //keeps track of where even numbers start in array
    for (int i = 0; i < nums.length; i++) {
        if (nums[i] % 2 != 0) { //if number is odd
            int temp = nums[evenindex];
            nums[evenindex] = nums[i];
            nums[i] = temp;
            evenindex++;
        }
    }
}
```
function oddEvenArray(nums) {
    if (nums.length <= 1) return nums;
    var odd = [], even = [];
    for (var i = 0; i < nums.length; i++) {
        if (i % 2 === 0) even.push(nums[i]);
        else odd.push(nums[i]);
    }
    return odd.concat(even);
}

Scoring:
+2: function header (return type, name, parameters)
+2: uses loop correctly
+3: places odd elements at front of array (any order)
+3: places even elements after odd elements (any order)

Most common error: Trying to work only one element ahead (for example, looking to see if nums[i+1] was out of order and then if it wasn’t, looking at nums[i+2]). Not only does this lead to indexing problems, but you’ll always be able to construct an array that this algorithm won’t work for.
9. You are given a 2D Java array which stores 24 temperature readings (1 per hour) for each day of the year:

```java
double[] temps = new double[365][24];
```

The first reading each day is taken at midnight, the next at 1am, the next at 2am, and so forth up until 11pm for a total of 24 readings. Write code which performs the following calculations. Note that you do not have to write a program or method. A snippet of code which mathematically calculates the result is sufficient.

(a) (5 points) Calculate the average temperature at 3pm for the year.

**Solution:**

```java
double sum = 0.0
for (int i = 0; i < temps.length; i++) {
    sum += temps[i][15];
}
double avg = sum/365;
```

Scoring:
+1 datatypes
+1 loop used correctly
+1 correct index for 3pm
+2 average calculated correctly

(b) (7 points) Calculate the average temperature for each day of the year and store those values into a new array.

**Solution:**

```java
double[] avgs = new double[365];

for(int i = 0; i < temps.length; i++) {
    double sum = 0.0;
    for (int j = 0; j < temps[i].length; j++) {
        sum += temps[i][j];
    }
    avgs[i] = sum /24.0;
}
```

Scoring:
+1 datatypes +2 loops correctly (nested loops)
+2 calcs daily average correctly
+2 stores average into new array correctly

Most common errors (on both parts): messing up data types (ints vs. doubles)

Another very common error was not paying attention to where your variables
were declared and initialized. For example, on part b using code like:

double sum = 0.0;
for (int i...) {
  for (int j...) {
    sum += temps[i][j];
  }
  avgs[i] = sum/24.0;
}

This code creates a running total of all (365*24) temperatures in the 2D array because the variable sum is never reset to to 0. In other words, it is not “reset” at the beginning of each day.

Another common error was initializing a variable in the wrong part of the loop. Example:

for (int i...) {
  int sum += temps[i][15];
}

Code like this will re-declare the variable sum every time the loop executes! Thus, the variable sum will not increment in the way the programmer had planned and will always be equal to the data stored in temps[364][15].