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 8 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>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points</td>
<td>8</td>
<td>9</td>
<td>14</td>
<td>9</td>
<td>12</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>72</td>
</tr>
</tbody>
</table>

Score: ___
1. (8 points) Evaluate each expression below. If the expression would result in a syntax or runtime error, give a brief description of the error. Otherwise, give the result of the expression.

```java
String s1 = "Exam 2";
int[] array1 = {0, 1, 2, 3, 4, 5, 6, 7};
String[] array2 = {"CS", "170", "Fall14"};
```

<table>
<thead>
<tr>
<th>Statement</th>
<th>Error or Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>array1[array1.length]</td>
<td>Error: index out of bounds</td>
</tr>
<tr>
<td>array1[s1.length]</td>
<td>Error: s1 is a String so need to use s1.length()</td>
</tr>
<tr>
<td>array1[s1.length()]</td>
<td>6</td>
</tr>
<tr>
<td>array1[s1.length()-1]</td>
<td>5</td>
</tr>
<tr>
<td>s1.charAt(array1[2])</td>
<td>a</td>
</tr>
<tr>
<td>s1.charAt(array2.length)</td>
<td>m</td>
</tr>
<tr>
<td>array1[(int)s1.charAt(0)]</td>
<td>Error: index out of bounds</td>
</tr>
<tr>
<td>array2[2].charAt(array1[s1.length()-3])</td>
<td>l</td>
</tr>
</tbody>
</table>

ASCII values:
- 'a' - 'z': 97 - 122
- 'A' - 'Z': 65 - 90
- '0' - '9': 48 - 57
- ' ' (space): 32
2. Sort each of the arrays below using the given algorithms. Show the array as each element is put into place. The code for each algorithm is given on the last page of the exam (with a comment showing where to draw the array), if you need it. Do not omit rows when the sorting algorithm does not make changes to the array.

(a) (3 points) \{9, -1, 14, 7, 4\}, Selection Sort

<table>
<thead>
<tr>
<th>Solution:</th>
<th>-1, 9, 14, 7, 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-1, 4, 14, 7, 9</td>
</tr>
<tr>
<td></td>
<td>-1, 4, 7, 14, 9</td>
</tr>
<tr>
<td></td>
<td>-1, 4, 7, 9, 14</td>
</tr>
</tbody>
</table>

(b) (3 points) \{14, 9, -1, 8, 10\}, Insertion Sort

<table>
<thead>
<tr>
<th>Solution:</th>
<th>9, 14, -1, 8, 10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-1, 9, 14, 8, 10</td>
</tr>
<tr>
<td></td>
<td>-1, 8, 9, 14, 10</td>
</tr>
<tr>
<td></td>
<td>-1, 8, 9, 10, 14</td>
</tr>
</tbody>
</table>

(c) (3 points) \{19, 21, 10, 3, 0\}, Bubble Sort

<table>
<thead>
<tr>
<th>Solution:</th>
<th>19, 10, 3, 0, 21</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10, 3, 0, 19, 21</td>
</tr>
<tr>
<td></td>
<td>3, 0, 10, 19, 21</td>
</tr>
<tr>
<td></td>
<td>0, 3, 10, 19, 21</td>
</tr>
</tbody>
</table>
3. (14 points) Determine what would be printed by the code below. If a variable cannot be printed at a given location, simply write “scoping error” as the output of that print statement

```java
public class Exam2 {
    public static int y = 10;
    public static int par = 15;

    public static int method1(int par) {
        System.out.println("2: " + par);
        System.out.println("3: " + y);
        System.out.println("4: " + Exam2.y);
        int y = 4;
        System.out.println("5: " + y);
        System.out.println("6: " + Exam2.y);
        par = -1;
        return par;
    }

    public static void main(String[] args) {
        int z = 15;
        String str = "CS170";
        System.out.println("1: " + z);
        y = method1(z);
        System.out.println("7: " + z);
        System.out.println("8: " + y);
        System.out.println("9: " + par);
        if (y <= 1) {
            String s1 = "CS";
            boolean y = false;
            if (Exam2.y <= 5) {
                String s2 = "170";
                System.out.println("10: " + str);
                System.out.println("11: " + s1);
                str = "CS170005";
            }
            System.out.println("12: " + str);
            System.out.println("13: " + s2);
            System.out.println("14: " + y);
        }
    }
}
```

**Solution:**
1: 15
2: 15
3: 10
4: 10
5: 4
6: 10
7: 15
8: -1
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>CS170</td>
</tr>
<tr>
<td>11</td>
<td>CS</td>
</tr>
<tr>
<td>12</td>
<td>CS170005</td>
</tr>
<tr>
<td>13</td>
<td>Scoping error</td>
</tr>
<tr>
<td>14</td>
<td>false</td>
</tr>
</tbody>
</table>
4. Consider the function `songs` defined below.

```java
public static void songs(String money, String peace, String love) {
    System.out.println("What the world needs now is " + love);
    System.out.println("The more " + money + " we come across");
    System.out.println("All I want is to have my " + peace + " of mind");
}
```

(a) (3 points) If you are given the following `main` method, what is the output of the code?
```
public static void main(String[] args) {
    String peace = "money";
    String love = "peace";
    String money = "love";

    songs(love, money, peace);
}
```

**Solution:**
Same as invoking function `songs("peace", "love", "money")`. So output is:
- What the world needs now is money
- The more peace we come across
- All I want is to have my love of mind.

(b) (3 points) The program is invoked with the command:
```
>>> java ProgramName peace money peace love 5
```
If you are given the following `main` method, what is the output of the code?
```
public static void main(String[] args) {
    songs(args[3], args[1], args[0]);
}
```

**Solution:**
Same as invoking method with `songs("love", "money", "peace")`. So output is:
- What the world needs now is peace
- The more love we come across
- All I want is to have my money of mind.

(c) (3 points) Given the function `songs` defined above and the 3 variables defined and initialized as follows:
```
String peace = "money";
String love = "peace";
String money = "love";
```
write a function call that would generate the output:
- What the world needs now is love
- The more money we come across
- All I want is to have my peace of mind

**Solution:**
To get desired output, call would need to be `songs("money", "peace", "love")` which leads to variable usage of:
- `songs(peace, love, money)`
5. For each part below, give the output of the code.

(a) (4 points)

```java
public static int method1(int x, String y) {
    if (y.length() >= 5) {
        System.out.println(y);
        return x + 10;
    }
    System.out.println(x);
    return x - 10;
}
```

```java
public static void main(String[] args) {
    int x = 15;
    System.out.println(method1(x, "Hello"));
    System.out.println(method1(x, "170"));
}
```

**Solution:**

```
Hello
25
15
5
```

(b) (2 points)

```java
public static void method1(int[] array) {
    for (int i = 0; i < array.length; i++) {
        array[i] = array[i] + 2;
    }
}
```

```java
public static void main(String[] args) {
    int[] array1 = {13, 14, 15, 16, 17};
    int[] array2 = {0, 1};
    array2 = array1;
    method1(array1);
    System.out.println(Arrays.toString(array1));
    System.out.println(Arrays.toString(array2));
}
```

**Solution:**

```
[15, 16, 17, 18, 19]
[15, 16, 17, 18, 19]
```
(c) (2 points)
   public static int[] method1(int[][] matrix) {
      int[] results = new int[2];
      results[0] = matrix.length;
      results[1] = matrix[1].length;
      return results;
   }
   public static void main(String[] args) {
      int[][] matrix = {{1, 2, 3}, {5, 6, 7, 8}, {9, 0}};
      int[] r = method1(matrix);
      System.out.println(r[0]);
      System.out.println(r[1]);
   }

   Solution: 3, 4

(d) (4 points)
   public static boolean method1(int x, int y) {
      System.out.println("integer");
      return true;
   }
   public static boolean method1(double x, double y) {
      System.out.println("double");
      return false;
   }
   public static void main(String[] args) {
      System.out.println( method1(4.0, 9) );
      System.out.println( method1(4, 9) );
   }

   Solution: double
   false
   integer
   true
6. (4 points) Write a method named `bodyMassIndex` which calculates a person’s BMI. The function should accept two doubles as parameters. One represents the weight in kilograms and the other represents the height in meters. Your function should return the double result as calculated by:

\[ BMI = \frac{\text{Weight}}{\text{Height}^2} \]

If the weight or the height is 0 or a negative number, the function should return -1.0.

**Solution:**

```java
public static double bodyMassIndex(double w, double h) {
    if (weight <= 0 || height <= 0) {
        return -1.0;
    } else {
        return w/(height*height);
    }
}
```

**Scoring:**
+1 method header
+1 calculation correct
+1 error detection
+1 return statement
7. (7 points) Write a method named `sumTwoSmallest` which takes an array of integers as input. The function should return the sum of the two smallest elements in the array. You may assume the array has at least 2 elements in it. Examples:

- `sumTwoSmallest({3, 2, 2, 4, 5})` returns 4 (sum of 2+2)
- `sumTwoSmallest({5, 4, 1, 2, 3})` returns 3 (sum of 1+2)

Solution:

```java
public static int sumTwoSmallest(int[] array) {
    int smallest;
    int nextSmallest;

    if (array[0] > array[1]) {
        smallest = array[1];
        nextSmallest = array[0];
    } else {
        smallest = array[0];
        nextSmallest = array[1];
    }

    for (int i = 2; i < array.length; i++) {
        if (array[i] < smallest) {
            nextSmallest = smallest;
            smallest = array[i];
        } else if (array[i] < nextSmallest) {
            nextSmallest = array[i];
        }
    }

    return smallest + nextSmallest;
}
```

Scoring: +1 method header
+2 smallest value identified
+2 2nd smallest value identified
+1 calculates sum of numbers
+1 return stmt
8. (9 points) Assume you are given a method, `isSortedDecreasing`, which takes an array of integers as input and returns `true` if the array is sorted in decreasing order (that is, from greatest to least). Otherwise, the method returns `false`.

Write a method `isMatrixDecreasing` which takes a 2D array of integers as input and returns `true` if all of the rows **AND** all the columns are in decreasing order. Otherwise it should return false.

```
9  8  7
6  5  4
3  2  1
```

For example, the matrix
```
9  8  7
6  5  4
3  2  1
```
would return `true`. However, the matrix
```
9  4  3  2
6  5  2  1
3  2  1  0
```
would return `false` as the 2nd column is not in decreasing order.

You may invoke/use the method `isSortedDecreasing`. You may assume the input matrix is not “ragged” (that all the rows are of equal length) and you may assume there is at least 1 element in the 2D array.

**Solution:**

```java
public static boolean isMatrixDecreasing(int[][] matrix) {
    // check each row by using function above
    for(int i = 0; i < matrix.length; i++) {
        if (!isSortedDecreasing(matrix[i])) {
            return false;
        }
    }

    // if we've reached this portion, all rows are sorted. now
    // check columns
    for(int i = 0; i < matrix[0].length; i++) {
        int[] a = new int[matrix.length];
        for (int j = 0; j < matrix.length; j++) {
            a[j] = matrix[i][j];
        }
        if (!isSorted(a)) {
            return false;
        }
    }
    return true;
}
```

Scoring:
+1 method header
+2.5 checks all rows
+4.5 checks all columns
+1 returns boolean value
Sorting Algorithms for an array, a.

- **Insertion Sort**
  ```java
  for(int i = 1; i < a.length; i++) {
    int currentElement = a[i];
    int k;
    for (k = i-1; k >= 0 && a[k] > currentElement; k--) {
      a[k+1] = a[k];
    }
    a[k+1] = currentElement;
  //Draw array here
  }
  ```

- **Bubble Sort**
  ```java
  int to_do = a.length-1;
  boolean did_swap = true;
  while (to_do != 0 & did_swap) {
    did_swap = false;
    for(int i = 0; i < to_do; i++) {
      if(a[i] > a[i+1]) {
        did_swap = true;
        int temp = a[i];
        a[i] = a[i+1];
        a[i+1] = temp;
      }
    }
    to_do = to_do - 1;
    //Draw array here
  }
  ```

- **Selection Sort**
  ```java
  for(int i = 0; i < a.length; i++) {
    int currentMin = a[i];
    int currentMinIndex = i;
    for (int j = i+1; j < a.length; j++) {
      if (currentMin > a[j]) {
        currentMin = a[j];
        currentMinIndex = j;
      }
    }
    if (currentMinIndex != i) {
      a[currentMinIndex] = a[i];
      a[i] = currentMin;
    }
    //Draw array here
  }
  ```