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>13</td>
<td>7</td>
<td>5</td>
<td>19</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>13</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>
</tr>
</tbody>
</table>
1. (13 points) Evaluate each expression. Then give the result of the evaluation and the data type of the result. If the expression cannot be evaluated or is not proper Java syntax, you may simply write “error” for the value. The first row has been done for you.

```java
String s1 = "CS170", s2 = "5", s3 = "13.0";
char c1 = '!', c2 = 'r';
int i1 = 4, i2 = 6; i3 = 315
double d1 = 3.0, d2 = 13.0; boolean b1 = false; b2 = true; b3 = false;
```

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4+1</td>
<td>5</td>
<td>int</td>
</tr>
<tr>
<td>i3 % 100</td>
<td>15</td>
<td>int</td>
</tr>
<tr>
<td>&quot;&quot; + i1 + i2</td>
<td>&quot;46&quot;</td>
<td>String</td>
</tr>
<tr>
<td>i2 / d1</td>
<td>2.0</td>
<td>double</td>
</tr>
<tr>
<td>i1 &gt;= d1 + d2 - 10</td>
<td>false</td>
<td>boolean</td>
</tr>
<tr>
<td>c2 + 2</td>
<td>116</td>
<td>int</td>
</tr>
<tr>
<td>(int)d2 + d1 + i3</td>
<td>331.0</td>
<td>double</td>
</tr>
<tr>
<td>b1</td>
<td></td>
<td>!(i3 &gt; d2 * d1)</td>
</tr>
<tr>
<td>b1 &amp;&amp; !b1</td>
<td>false</td>
<td>boolean</td>
</tr>
<tr>
<td>i2 + 6 &lt; i2 + 7 &lt; i2 + 8</td>
<td>error</td>
<td>can’t compare booleans to ints</td>
</tr>
<tr>
<td>(int)s1.charAt(3)</td>
<td>55</td>
<td>int</td>
</tr>
<tr>
<td>-(-i2)</td>
<td>6</td>
<td>int</td>
</tr>
<tr>
<td>!(b1 &amp;&amp; (b2</td>
<td></td>
<td>!b3))</td>
</tr>
<tr>
<td>d1 + c1 + s2</td>
<td>&quot;36.05&quot;</td>
<td>String</td>
</tr>
</tbody>
</table>

Notes:
- The parenthesis operator has the highest precedence of all operators.
- The logical not operator and the casting operator have higher precedence than arithmetic operators, relational/comparison operators, and the other logical operators.
- Arithmetic operators have higher precedence than relational/comparison or logical operators.
- Relational/comparison operators have higher precedence than logical operators.
- Assignment operators have the lowest precedence of all operators.
2. Briefly (no more than 2 sentences!) answer each question below. You may give an example if you wish.

(a) (1 point) What is the function of a compiler?

Solution: translate high level language (such as Java) to a low level (machine) language.

(b) (1 point) What is the command used to compile a file named MyProgram.java?

Solution: javac MyProgram.java

(c) (1 point) In Java, what is a library?

Solution: container for related classes

(d) (1 point) What is casting?

Solution: Changing datatype of a value. Can be explicit (done by programmer using casting operator) or implicit (handled by Java in “safe” situations).

(e) (1 point) List 3 comparison operators.

Solution: >=, >, <=, <, ==, !=

(f) (1 point) List 3 shortcut assignment operators.

Solution: +=, -=, *=, /=, %=, ++, --

(g) (1 point) What is the smallest unit of executable program code in Java?

Solution: a statement
3. (5 points) Assume the statements below are part of a Java program which compiles and runs. What is the output if the user types 7?

```java
Scanner in = new Scanner(System.in);
int num = in.nextInt();

if (num < 0 && num > 10) {
    System.out.println("Bert");
} else if (num < 0 || num > 10) {
    System.out.println("Ernie");
} else {
    System.out.println("Big Bird");
}

System.out.println("One. Ah-ah-ah");

if (num > 0 && num < 10) {
    System.out.println("Oscar");
} else if (num >= 0 && num <= 10) {
    System.out.println("Grover");
} else if (num >= 5 || num <= 8) {
    System.out.println("Cookie Monster");
} else {
    System.out.println("Elmo");
}

System.out.println("Two. Ah-ah-ah");

if (num >= 4) { System.out.println("Telly"); if (num > 5) {
    System.out.println("Baby Bear"); } else { System.out.println("The Count")}
```

**Solution:** Big Bird
One. Ah-ah-ah.
Oscar
Grover
Two. Ah-ah-ah.
Telly
Baby Bear

Scoring: -.5 for each incorrect inclusion or exclusion except “One...” and “Two...”
4. For each of the code fragments below, give the output. If the code contains a syntax error, correct it and then give the output of the corrected code. If the code results in an infinite loop, write the first few outputs, and then indicate that it is an infinite loop.

(a) (3 points)
```java
for (int i = 1; i < 15; i+=1) {
    if (i % 6 == 5) {
        break;
    } else {
        i = i + 1;
        System.out.println(i);
    }
}
```

Solution: 2
4

(b) (3 points)
```java
String s = "Emory";
for(int i = s.length()-2; i>=0; i--) {
    System.out.print(s.charAt(i+1));
}
```

Solution: yrom

(c) (3 points)
```java
for (int i = -2; i < 2; i++) {
    System.out.println(i*2);
}
```

Solution: -4, -2, 0, 2
(d) (3 points)
String s = "Emory";
for(int i = 0; i < s.length(); i++) {
    s = s + s.charAt(i);
    System.out.println(s)
}

Solution: infinite loop! The string s grows with every iteration of the loop, so s.length() increases by 1 each iteration of the loop. Many students thought the loop ended when the string reached “EmoryEmory”.
EmoryE
EmoryEm
EmoryEmo
...

(e) (2 points)
int n = 0;
while(n < 4); {  
    System.out.println(n);
    n++;
}

Solution: This is tricky! Notice the ; after the while. As written, this is an infinite loop that always prints 0. I also accepted answers where the semicolon was removed and the output was 0, 1, 2, 3. However, if you did this, you had to indicate that you knew/recognized that the semi-colon changed the execution.

(f) (3 points)
int sum = 0;
for(int i = 0; i <= 3; i++)
    sum+= i;
System.out.println(sum);

Solution: Careful! The SOP lines looks like it’s included in the for loop, but the for loop was written without curly braces! (This was very similar to one of your earlier quiz problems.) Output is only the final sum after the loop has finished: 6. Also accepted if students added braces to match the indentation in which case the output was 0, 1, 3, 6.

(g) (2 points)
String name = "456";
if( name.length() = 3) {
    System.out.println("Length is three.");
}

Solution: Syntax error. Assignment operator (=)used instead of equality operator (==). After correcting error, output is “Length is three”
5. Professor Summet wants to swap the values of the two variables x and y. In other words, after the code below executes, x should have the value 13 and y should have the value 14. However, her code doesn't work!

    /****************************************/
    int x = 14;
    int y = 13;
    System.out.println("x: " + x);
    System.out.println("y: " + y);

    //swap the values of x and y
    x = y;
    y = x;
    System.out.println("after swap x: " + x);
    System.out.println("after swap y: " + y);
    /****************************************/

(a) (2 points) What is the output of the code above as it is currently written?

Solution:

    x: 14
    y: 13
    after swap x: 13
    after swap y: 13

(b) (3 points) Rewrite the swap code (after the comment in the original code) so that the values are correctly swapped. Hint: you may add additional variables if you need to.

Solution:

    int temp = x;
    x = y;
    y = temp;

Common error: writing a solution that only worked with the given values of x and y. For example: x = 14; y = 13;
6. Professor Summet wrote the code below to check whether the times for two appointments overlap given 4 integer values for the variables start1, end1, start2, and end2. The starting times of the two appointments are represented by start1 and start2 while end1 and end2 represent the respective ending times. For simplicity, assume appointments are given in 24-hour (i.e., military) time (0-24) and start on the hour. Consider the following code to solve this problem:

```
int start1, end1, start2, end2, s, e;
//variables initialized here with values in table below

if (start1 > start2) {
    s = start1;
} else {
    s = start2;
}

if (end1 < end2) {
    e = end1;
} else {
    e = end2;
}

boolean overlap = s < e;
```

(a) (3 points) Complete the table below with the value that results given the 4 variable values for start1, end1, start2, end2.

<table>
<thead>
<tr>
<th>start1</th>
<th>end1</th>
<th>start2</th>
<th>end2</th>
<th>overlap</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>12</td>
<td>11</td>
<td>13</td>
<td>true</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>11</td>
<td>12</td>
<td>false</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>14</td>
<td>15</td>
<td>false</td>
</tr>
</tbody>
</table>

(b) (2 points) Does the code work as intended? If not, explain why not.

**Solution:** Yes, it does, assuming that no appointments span a day (e.g., begin at 20:00 and end at 1:00). Some students also modified the code to not allow appointments to start and end at the same time (e.g., Appt1 end at 12:00 and Appt2 start at 12:00).
7. (8 points) Complete the code below which determines if 3 integer numbers are “evenly distributed.” “Evenly distributed” numbers are separated by the same amount. In other words, the difference between the largest and middle number is the same as the difference between the middle and the smallest number. Your code should print out a statement to indicate whether or not the numbers are evenly distributed.

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>Evenly Distributed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>9</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>9</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>6</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>9</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>6</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>6</td>
<td>No</td>
</tr>
</tbody>
</table>

Hint: recall that the Math.max function takes two integers as input parameters and returns the larger of the two. Likewise, the Math.min function takes two integers but returns the smallest of them.

```java
int a, b, c;

//You may assume the variables are initialized here. You DO NOT need
//to initialize them or use a Scanner to initialize them.

/*************** Your code here *****************/

Solution:

```java
int big, mid, small;
if (a >= b && a >= c) {
    big = a;
    mid = Math.max(b, c);
    small = Math.min(b, c);
} else if (b >= a && b >= c) {
    big = b;
    mid = Math.max(a, c);
    small = Math.min(a, c);
} else {
    big = c;
    mid = Math.max(a, b);
    small = Math.min(a, b);
}

if (big - mid == mid - small)
    System.out.println("Evenly distributed");
else
    System.out.println("Not evenly distributed");
```

+2 Identified overall max
+2 Identified overall min
+2 Identified middle number
+2 Correctly determines distributed or not and prints to terminal

Common mistakes:
• Not accounting for case where all numbers were equal.

• Accounting for only certain cases (e.g. code only works when a is the largest number or c is the smallest, etc).
8. (13 points) Complete the program below which determines whether or not a phone number is “valid”. A phone number is considered to be valid if it is a 7 digit number which does not contain the number 15 (ie the digits ‘1’ and ‘5’ in order) as any part of it. You may assume the user always enters a 7 digit integer.

import java.util.Scanner;

public class ValidNum{
    public static void main (String[] args){
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = in.nextInt();
        /*----------- Your code here -------------*/
    }
}

Solution:

boolean valid = true;
while(num != 0) {
    int rem = num % 100; // check last 2 digits!
    if (rem == 15) {
        valid = false;
        break; //not necessary but more efficient
    }
    num = num / 10; // remove only 1 digit!
}

if (valid) { // or if(valid == true)
    SOP("Number is valid");
} else {
    SOP("Number is invalid");
}

Another solution using a conversion to a String:

String s = "" + num;
boolean valid = true;
for (i = 0; i < s.length()-1; i++) {
    if (s.charAt(i) == '1' & s.charAt(i+1)=='5') {
        valid = false;
    }
}

if (valid) {
    S.O.P("valid");
} else {
    S.O.P("invalid");
}
Scoring:
+3 loop runs correctly
+2 inspects 2 digits of loop each time +2 correctly identifies 15 in number
+2 correctly identifies valid numbers
+2 correctly identifies invalid numbers
+2 prints output (valid or invalid) to screen correctly

Common mistakes:

- Printing “valid” or “invalid” each iteration of loop rather than just printing once at end of loop.

- Converting to String by casting. (ie `String s = (String) num;`) You can only use the casting operator for primitive datatypes.

- Letting loop go too far and trying to examine characters that don’t exist:

```java
for (int i = 0; i < s.length(); i++) {
    if (s.charAt(i) == '1' && s.charAt(i+1) == '5') ...
}
```

Note that `s.length()` will be 7 so loop will run from 0 to 6 (inclusive). When i is 6, you will be examining characters at indices 6 and 7 (which doesn’t exist) and you will generate an IndexOutOfBoundsException runtime error.