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 13 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>12</td>
<td>24</td>
<td>12</td>
<td>12</td>
<td>10</td>
<td>12</td>
<td>18</td>
<td>100</td>
</tr>
</tbody>
</table>

Score: ______________________________
1. (12 points) Definitions:
Please give a very basic (brief) definitions for 4 of the 5 following terms. You may include an example if it is helpful to your answer. Please mark the one that you do not want to answer; otherwise, I will assume that the last term is to be discarded.

- **Safe conversion**
  A safe conversion is a type conversion from a "smaller" data type to a "larger" data type (a data type with a range larger and containing the smaller data type in it).

- **Shortcut (shorthand) operators**
  Shortcut (shorthand) operators are shorter ways to express something that is already available in the Java programming language. Examples of shortcut operators are +=, -=, *= and etc.

- **Statement**
  A statement is the smallest unit of execution in Java. It is usually terminated by a semi-colon.

- **The **continue** statement**
  The **continue** statement is used in loop-statement to skip all following statements till the end of the loop body and resume the control flow at the loop header. For while loop, the program immediately goes back to the loop condition check.

- **Dangling else ambiguity**
  The dangling **else** ambiguity occurs when an else-part of a statement might be associated with more than one if-statement, usually in a nested if-else-statements, where block statements (curly braces) are not used to delimit the boundary of each if-statement component.

**Solution:** 3 points for each definition.
2. (24 points) Evaluate the following expressions. Fill in the value of the evaluation and the data type of the result. The first row has been completed as an example. Assume that expressions are evaluated independently (i.e. the values of all variables are reset to the default values below at the beginning of each expression). If the expression has a syntax error, write "Error". You can refer to the ASCII table on the last page of the exam.

```java
int x = -10;
float f = 2.0f;
double d = 5.0;
char letter = 'b';
String str1 = "-10", str2 = "Hello";
```

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>x + 1</td>
<td>-9</td>
<td>int</td>
</tr>
<tr>
<td>d * 2 + (x + 15) / 8</td>
<td>10.0</td>
<td>double</td>
</tr>
<tr>
<td>d % 2 + str2.substring(1, 4) + d / 2</td>
<td>1.0ell2.5</td>
<td>String</td>
</tr>
<tr>
<td>str1.charAt(1) - 2 % 8</td>
<td>47</td>
<td>int</td>
</tr>
<tr>
<td>(char) '3' + 2.0 == '5'</td>
<td>true</td>
<td>boolean</td>
</tr>
<tr>
<td>(int) str1.substring(1,2) + f</td>
<td>Error</td>
<td>Error</td>
</tr>
<tr>
<td>'x'&gt; letter &gt;= 'a'</td>
<td>Error</td>
<td>Error</td>
</tr>
<tr>
<td>str2.substring(1,3) + 'n' + 2</td>
<td>eln2</td>
<td>String</td>
</tr>
<tr>
<td>Double.parseDouble(str1) / 4</td>
<td>-2.5</td>
<td>double</td>
</tr>
<tr>
<td>(true</td>
<td></td>
<td>x == d) &amp;&amp; letter == 'A'</td>
</tr>
<tr>
<td>f++ * --d + x</td>
<td>-2.0</td>
<td>double</td>
</tr>
<tr>
<td>(str1 + &quot;\n\t\n&quot;).length()</td>
<td>7</td>
<td>int</td>
</tr>
<tr>
<td>(x = 3) &lt; 5 &amp;&amp; !(f &lt; 3)</td>
<td>false</td>
<td>boolean</td>
</tr>
<tr>
<td>Solution:</td>
<td>For every problem, 1 point for the correct value and 1 point for the correct data type.</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If there is partial work and only a very minor error: 0.5 point.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the actual answer was an Error, but a value and type was written instead: 0 point.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the actual answer has a value and a type, but &quot;Error&quot; was written instead: 0 point.</td>
<td></td>
</tr>
</tbody>
</table>
3. (12 points) Suppose you are given the following variable definitions:

```java
int i = 2;
double d = 4.0;
```

write the values that will be printed to the console (terminal). If a statement has syntax errors, write Syntax Error, briefly describe the error(s) and provide the reason (or a way to correct the error(s)). If there is an infinite loop, write "Infinite loop". Assume all code fragments are independent. The variables i and d reset the to the above values at the beginning of each code fragment. You can refer to the ASCII table on the last page of the exam.

a) Code:

```java
System.out.println((char) 'd' + d);
System.out.println(d++ + i + "i" + i * 2);
```

Output:
104.0
6.0i4

b) Code:

```java
double z = 6++ - 7 + a;
int u = (int) z * 2;
if (z > 0) {
    System.out.print(z + 2);
}
```

Syntax errors
1) 6++ is not a valid operation or literal.
2) Variables a and z have not been declared or initialized

(c) Code:

```java
while (i > 0) {
    System.out.println("Morning");
} still if (i / 2 == 0) {
    System.out.println("Evening");
    i = (char) i;
    System.out.println("Night");
}
```

Error:
Syntax error: There is no "still" keyword in java.

```
d) Code:

while (i < 0 || false);
```


if (i % 2 == 0) {
    System.out.print(i);
} 

i = i * 2;

System.out.print(i);

Output:
24

(No infinite loop here.)

e) Code:

String s = "canvas";
int count = s.length() - 3;
while (count < s.length() ) {
    if (s.charAt(count) < 'a') {
       count++;
    }
}

Infinite loop. Variable count stays at value 3.

f) Code:

x = 10;
if ( x <= 20 ) {
    System.out.println(x + 3);
} if (x > 0) {
    System.out.println(x * 3);
} if (x * 2 > 4) {
    System.out.println(x == 5);
}

Syntax error: The type of variable x is not determined.

Output (assuming x is of type int):
13
30
false

Solution: Each problem: 2 points.
4. (12 points) Assume the statements below are part of a Java program.

```java
Scanner in = new Scanner(System.in);
int x = in.nextInt();
char ch = 'b';
if (x > 3.0 || x - 10 < 0){
    System.out.println("A");
    if(x > 20){
        if(true && !false){
            System.out.println("B");
            ch++;
        }
        System.out.println("C");
    } else {
        System.out.println("D");
    }
    System.out.println("E");
}
System.out.println("x = " + x);
System.out.println("ch = " + ch);
switch(ch){
    case 'c':
        System.out.println("P");
    case 'b':
        System.out.println("Q");
    case 'a':
        System.out.println('R');
        break;
    case 'x':
        System.out.println("S");
        break;
    default:
        System.out.println("Default");
}
x = 10;
System.out.println(x +"\"Z"\"U");
```

Assume the code compiles.

What is the output on the terminal if the user types in 100?
What is the output on the terminal the user types in 6?

**Solution:**

Part a) Output:

```
A
B
C
E
x = 100
ch = c
P
Q
```
Part b:

A
D
E
\( x = 6 \)
ch = b
Q
R
\( 10^\text{Z"U} \)
5. (10 points) The following code is a Java program that compiles and runs. Please provide the output of the program (i.e. what values are printed to the console/terminal).

```java
public class LoopAndSwitch {
    public static void main(String[] args) {
        int a = 1;
        int b = 2;

        while (a < 10) {
            switch (a) {
                case 1:
                case 2:
                    System.out.println(a);
                    a++;
                    break;
                case 4:
                    System.out.println(a);
                    a += b;
                    break;
                case 5:
                    System.out.println(a - b);
                    break;
            }
            if (a % 7 == 0) {
                System.out.println(a);
                break;
            }
            a++;
        }
        System.out.println("a = " + a);
        System.out.println("b = " + b);
    }
}
```

Solution:

Output:

1
4
5
7
a = 7
b = 2
6. (12 points) Complete the program below. The program should keep prompting the user to enter an integer, one at a time. The program should keep doing so until the user enters 0. Then the program should output the sum of all numbers that the user has entered and their arithmetic mean (exclude the last number 0 that the user enters to stop the input prompt).
Sample output:

Please enter an integer: 5
Please enter an integer: 3
Please enter an integer: -3
Please enter an integer: 1
Please enter an integer: 0
The sum is 6.0
The mean is 1.5

Note: The mean is 1.5, because the sum is 6 and there are 4 numbers not including 0.

--- Solution ---

```java
import java.util.Scanner;
public class FindSumAndMean {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

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

    }
}
```

```java
Solution:
import java.util.Scanner;
public class FindSumAndMean {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        double sum = 0;
        int count = 0;
        System.out.print("Please enter an integer: ");
        int n = in.nextInt();
        while (n != 0) {
            sum += n;
            count++;
            System.out.println("Please enter an integer: ");
            n = in.nextInt();
        }

        System.out.print("The sum is "+ sum);
        System.out.print("The mean is "+ sum / count);
    }
}
```
7. (18 points) Prime factors. Complete the following program:
   The program prompts the user to enter an integer and store it into the variable a. The program should
   output all prime factors of the number in a comma-separated sequence. A number if prime if and only
   if it is only divisible by 1 and itself.
   Hint: You should use the repeated-division of "some divisors" on number a to eventually reach 1.

<table>
<thead>
<tr>
<th>Sample Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>3, 7,</td>
</tr>
<tr>
<td>8</td>
<td>2, 2, 2,</td>
</tr>
<tr>
<td>60</td>
<td>2, 2, 3, 5,</td>
</tr>
<tr>
<td>200</td>
<td>2, 2, 5, 5,</td>
</tr>
<tr>
<td>17</td>
<td>17,</td>
</tr>
</tbody>
</table>

   import java.util.Scanner;
   public class PrimeFactors {
       public static void main(String[] args) {
           Scanner in = new Scanner(System.in);
           System.out.print("Please enter an integer: ");
           int a = in.nextInt();
           /*----------- Your code here -------------*/
       }
   }

   Solution:

   import java.util.Scanner;
   public class PrimeFactors {
       public static void main(String[] args) {
           Scanner in = new Scanner(System.in);
           System.out.print("Please enter an integer: ");
           int a = in.nextInt();

           int n = 2; // Potential factor

           while (a > 1 && n <= a) {
               if (a % n == 0) {
                   // We found a factor!
                   System.out.print(n + ", ");
                   a = a / n; // Find prime factors for the remaining quotient
               } else {
                   // This is not a factor so try a larger number
                   n++;
               }
           }
       }
   }
### ASCII Table

Note that lowercase: 97 <= x <= 122
Note that uppercase: 65 <= x <= 90

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>c</td>
<td>d</td>
<td>e</td>
<td>f</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
</tr>
<tr>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
</tr>
<tr>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
</tr>
<tr>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example:**
- Difference between A (65) and a (97) is 32!
- Difference between Q (81) and q (113) is 32!