CS170 (005): Introduction to Computer Science I
Exam 1

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.

Time: This exam has 5 questions on 7 pages, including this title page and an ASCII chart on the last page. Please make sure all pages are included. You will have 50 minutes to complete this exam.

Honor code: 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 : ______________________________
1) **(20 pts)** For each of the following terms give the definition of the term. You just need to give a good description of the term, a formal definition is not needed. You can use an example if it will help your explanation. Please keep your answers short.

a) Data type

A data type lets the computer know what type of encoding scheme to use when interpreting the number stored in a variable.

b) Boolean expression

A Boolean expression is an expression that evaluates to either true or false. Example: 
\[ i \geq 12 \]

c) Keyword

A keyword is a reserved word in a programming language used for a specific purpose.

d) Condition clause

The condition clause is the boolean expression part of a conditional statement that evaluates to true or false.

e) Escape sequence

An escape sequence is a combination of the escape character followed by another character. The escape sequence is used to represent characters that cannot be typed. Example: this escape sequence is used to represent the double quotes character in a string literal. Since the double quotes character itself denotes the beginning and end of a string literal, we can’t just type it by itself.
2) (30 pts) Evaluate each of the following expressions. Give the result and the data type in the space provided. If the expression can not be evaluated or is not proper Java syntax, write "error" as the result. The values and data types of all variables used in the expressions are given. The first one has been completed for you.

```java
int a = 5, b = 4, c = 10;
double d1 = 4.0, d2 = 8.0;
String s1 = "Java", s2 = "rocks", s3 = "20";
char m = 't';
```

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>a + 10</td>
<td>15</td>
<td>int</td>
</tr>
<tr>
<td>a + b * 2</td>
<td>13</td>
<td>int</td>
</tr>
<tr>
<td>c % b</td>
<td>2</td>
<td>int</td>
</tr>
<tr>
<td>a / b</td>
<td>1</td>
<td>int</td>
</tr>
<tr>
<td>(double)c</td>
<td>10.0</td>
<td>double</td>
</tr>
<tr>
<td>s1 * 2</td>
<td>Error</td>
<td></td>
</tr>
<tr>
<td>s2 + &quot;! &quot; + s1</td>
<td>rocks!java</td>
<td>String</td>
</tr>
<tr>
<td>a + c % 7</td>
<td>8</td>
<td>int</td>
</tr>
<tr>
<td>(int)(d1 / 0.5)</td>
<td>8</td>
<td>int</td>
</tr>
<tr>
<td>d1 &lt; 10.1 &amp; &amp; !(a == 5)</td>
<td>false</td>
<td>boolean</td>
</tr>
<tr>
<td>(char)(m + 2)</td>
<td>'v'</td>
<td>char</td>
</tr>
<tr>
<td>d1 % 5 + d2</td>
<td>Question dropped</td>
<td></td>
</tr>
<tr>
<td>(c - d2) * 5</td>
<td>10.0</td>
<td>double</td>
</tr>
<tr>
<td>s3 + &quot;13&quot;</td>
<td>&quot;2013&quot;</td>
<td>String</td>
</tr>
<tr>
<td>-a * (2 * b - - a)</td>
<td>-65</td>
<td>int</td>
</tr>
<tr>
<td>m - '6'</td>
<td>62</td>
<td>int</td>
</tr>
</tbody>
</table>
3) (20 pts) Given the following variables that have already been assigned some value.

```java
int a, b, c;

write a Java statement(s) that:

a) Prints the value of \(a \times c\).

System.out.println(a * c);

b) Prints the value of \(a\) if \(a\) is the smallest value.

if( a <= b && a <= c )
{
    System.out.println(a);
}

c) Prints the value of \(b\) if \(b\) is divisible by \(a\) but not by \(c\).

if( b % a == 0 && b % c != 0 )
{
    System.out.println(b);
}

d) Prints "Yes" if any of the variables are even and "No" if not.

if( a % 2 == 0 || b % 2 == 0 || c % 2 == 0 )
{
    System.out.println("Yes");
}
else
{
    System.out.println("No");
}
```
4) (15 pts) Assume the following statements are part of a valid Java program that compiles and runs. Give the output of the program if the user enters 20.

```java
int adj;
Scanner in = new Scanner(System.in);
System.out.print("Enter a number: ");
int grade = in.nextInt();
adj = (int)(grade * 1.5);

while( grade > 0 ) {
    System.out.print("*");
    grade = grade / 5;
}
System.out.println("Score: " + adj);

if( adj > 100 ) {
    System.out.println("Not possible!");
} else if( adj >= 90 ) {
    System.out.println("Good job");
} else if( adj < 80 ) {
    System.out.println("Not bad");
} else if( adj > 20 ) {
    System.out.println("Oh No!!");
} else {
    System.out.println("What went wrong?");
}
```

Output:

```
Enter a number:
**Score: 30
Not bad
```
5) (15 pts) The following Java program finds all the divisors for a given number and displays them. With the exception of 5. That is, if 5 is a divisor of the given number, don’t display it. Find the errors in the program, describe the error and how to fix it.

```java
import java.util.Scanner;

public class Div {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int num, x;

        System.out.println("Enter a number: ");
        num = in.nextFloat();  <-- Error 1
        x = 1;  <-- Error 2
        while( x <= num );  <-- Error 3
        {
            if( !(num % x) ) {
                continue;
            }
            System.out.println(x);
            x++;  <-- Error 4, 5 & 6
        }
    }
}
```

Error 1: Reading an int, should be in.nextInt()
Error 2: Assignment statements use one =
Error 3: Errant semicolon
Error 4: Condition doesn’t check if x is not a divisor properly  if( num % x != 0 )
Error 5: No check if divisor is 5  if( num % x != 0 || x == 5 )
Error 6: Missing x++; before continue statement causes an infinite loop
<table>
<thead>
<tr>
<th>Code</th>
<th>Character</th>
<th>ASCII Value</th>
<th>Decimal Value</th>
<th>Binary Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>nul</td>
<td>016</td>
<td>032</td>
<td>00000110</td>
</tr>
<tr>
<td>001</td>
<td>soh</td>
<td>017</td>
<td>033</td>
<td>0000011001</td>
</tr>
<tr>
<td>002</td>
<td>stx</td>
<td>018</td>
<td>034</td>
<td>0000011010</td>
</tr>
<tr>
<td>003</td>
<td>etx</td>
<td>019</td>
<td>035</td>
<td>0000011011</td>
</tr>
<tr>
<td>004</td>
<td>eot</td>
<td>020</td>
<td>036</td>
<td>0000011100</td>
</tr>
<tr>
<td>005</td>
<td>eum</td>
<td>021</td>
<td>037</td>
<td>0000011101</td>
</tr>
<tr>
<td>006</td>
<td>ack</td>
<td>022</td>
<td>038</td>
<td>0000011110</td>
</tr>
<tr>
<td>007</td>
<td>bel</td>
<td>023</td>
<td>039</td>
<td>0000011111</td>
</tr>
<tr>
<td>008</td>
<td>can</td>
<td>024</td>
<td>040</td>
<td>0000100000</td>
</tr>
<tr>
<td>009</td>
<td>tab</td>
<td>025</td>
<td>041</td>
<td>0000100001</td>
</tr>
<tr>
<td>010</td>
<td>lnf</td>
<td>026</td>
<td>042</td>
<td>0000101010</td>
</tr>
<tr>
<td>011</td>
<td>vrt</td>
<td>027</td>
<td>043</td>
<td>0000101011</td>
</tr>
<tr>
<td>012</td>
<td>np</td>
<td>028</td>
<td>044</td>
<td>0000101100</td>
</tr>
<tr>
<td>013</td>
<td>cr</td>
<td>029</td>
<td>045</td>
<td>0000101101</td>
</tr>
<tr>
<td>014</td>
<td>sp</td>
<td>030</td>
<td>046</td>
<td>0000101110</td>
</tr>
<tr>
<td>015</td>
<td>si</td>
<td>031</td>
<td>047</td>
<td>0000101111</td>
</tr>
</tbody>
</table>

Regular ASCII Chart (character codes 0 - 127)