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 11 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>16</td>
<td>6</td>
<td>16</td>
<td>14</td>
<td>12</td>
<td>18</td>
<td>18</td>
<td>100</td>
</tr>
<tr>
<td>Score:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. (16 points) Evaluation table:
Fill out the following table. Evaluate the Java expression in the first column and put the result in the second column. Assume that each expression is evaluated independently (means not in sequence). In the third column indicate the datatype of the result. The first row has been done for you. If the expression cannot be evaluated, you may write "error".

You have the following variables declared and initialized:

```java
boolean b = true;
double d = 8.0, e = 2.0;
int i = 5, j = 12;
char c = 'C';
String str1 = "CS170", str2 = "000";
```

<table>
<thead>
<tr>
<th>Java expression</th>
<th>Result</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>d + i</td>
<td>13.0</td>
<td>Double</td>
</tr>
<tr>
<td>str1.length() + str2.length()</td>
<td>8</td>
<td>int</td>
</tr>
<tr>
<td>(++i + j * 2) % 2</td>
<td>0</td>
<td>int</td>
</tr>
<tr>
<td>b &amp;&amp; ! (i &gt; j)</td>
<td>True</td>
<td>Boolean</td>
</tr>
<tr>
<td>str1 + c + &quot; fans!&quot;</td>
<td>&quot;CS170C fans!&quot;</td>
<td>String</td>
</tr>
<tr>
<td>i / j + d / e</td>
<td>4.0</td>
<td>Double</td>
</tr>
<tr>
<td>str1.substring(2, 5)</td>
<td>&quot;170&quot;</td>
<td>String</td>
</tr>
<tr>
<td>str2.charAt(1) == c</td>
<td>False</td>
<td>Boolean</td>
</tr>
<tr>
<td>Integer.parseInt(str2) + j</td>
<td>12</td>
<td>int</td>
</tr>
</tbody>
</table>
2. (6 points) Definitions
For each of the following give a basic definition of the term. You do not need to give a formal definition.

- **default** (give an example (a piece of code) in Java and explain it)

  **Solution:** The default section handles all values that are not explicitly handled by one of the case sections. Example:
  
  ```java
  int a = 5;
  switch (a){
  case 1:
    break;
  default:
    break;
  }
  ```

- **Casting**

  **Solution:** Casting a variable data type to some other data type. Example:
  
  ```java
  int x = 5;
  double y = (double) x;
  ```

- **Infinite loop**

  **Solution:** Execution of the loop which loops endlessly, either due to the loop having no terminating condition, having one that can never be met, or one that causes the loop to start over.

3. (16 points) Outputs and syntax problems
The following code snippets could have syntax problems (syntax problems mean that the code will not compile). For those which are correct, print the output of the code. For those with an error, indicate what the error is and correct the code.

- Snippet 1

  ```java
  int a = 50;
  if (a == 60){
    System.out.println("First if");
  }
  else if (a < 45 || a > 55){
    System.out.println("Second if");
  }
  else{
    System.out.println("Else");
  }
  ```
Output or error:

Solution: "Else"

• Snippet 2

1 int a = 10;
2 if (a = 5){
3 System.out.println("If");
4 }
5 else{
6 System.out.println("Else");
7 }

Output or error:

Solution: Operator "=" is assignment operator, which cannot be used in conditional expression. To compare operator "==" should be used. Fixed:

if (a == 5){

• Snippet 3

1 int i = 0;
2 while(i < 10){
3 System.out.println(i);
4 i += 2;
5 }

Output or error:

Solution: 0
2
4
6
8

• Snippet 4

1 int a = 30, c;
2 if (a == 10){
3 c = 1;
4 }
5 else if (a == 20){
6 c = 2;
7 }
8 else if (a > 20){
9 c = 3;
Output or **error**:  

<table>
<thead>
<tr>
<th>Solution:</th>
<th>else keyword does not take any conditions, it means &quot;otherwise&quot;. Two fixes exist:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>else {</td>
</tr>
<tr>
<td></td>
<td>or</td>
</tr>
<tr>
<td></td>
<td>else if (a &gt; 20){</td>
</tr>
</tbody>
</table>

```java
10 }  
11 System.out.println(c);
```
4. (14 points) Problem 4

The Java code below runs on the machine. What will the output be if the user types:

Enter a number: 10
Enter a word: CS17ixxx

```java
int userInt;
String word;

Scanner input = new Scanner(System.in);

System.out.print("Enter a number: ");
userInt = input.nextInt();
System.out.print("Enter a word: ");
word = input.next();

if ((userInt % 5) == 0) {
    if (word.charAt(0) == 'C') {
        if (word.charAt(1) == 'S') {
            System.out.println("We got a CS!");
        } else {
            System.out.println("Got some word: " + word);
        }
    } else {
        System.out.println(userInt + " "+ word.charAt(0));
    }
} else {
    System.out.println("Number: " + userInt + " is divisible by 5.");
}
else{
    System.out.println("Number: " + userInt + " is not divisible by 5.");
}

switch(userInt){
case 0:
    System.out.println("Magic number: " + userInt * 10 / 5);
    break;
case 1:
    if (word.charAt(0) == 'C' && word.charAt(1) == 'S') {
        System.out.println("We are CS class!");
    } else {
        System.out.println("Uhm, not CS!");
    }
    break;
case 10:
    System.out.println("We have 10 here!");
    break;
case 100:
    break;
}
```

Output:

**Solution:** We got a CS!
Number 10 is divisible by 5.
We have 10 here!
5. (12 points) Prefixes

Consider the code below with logical errors that should print metric prefixes for factors. For example, for 1000 it should print kilo:

Enter a factor: 1000
For factor: 1000, prefix is: kilo

```java
import java.util.Scanner;
public class Prefixes{
    public static void main(String [] args){
        int factor;
        String prefix;
        Scanner input = new Scanner(System.in);
        System.out.println("Enter a factor: ");
        factor = input.nextInt();
        switch (factor){
            case 10:
                prefix = "deca";
                break;
            case 100:
                prefix = "hecto";
                break;
            case 1000:
                prefix = "kilo";
                break;
            case 1000000:
                prefix = "mega";
                break;
            case 1000000000:
                prefix = "giga";
                break;
            default:
                prefix = "Unknown";
                break;
        }
        System.out.println("For factor: "+factor+" , prefix is: "+prefix);
    }
}
```

- What does the code output for input 1000.

Solution: "For factor: 1000, prefix is: giga"

- Explain the logical error in the code.

Solution: A few cases in switch statement miss break statements, therefore for some factor values (100, 1000, 1000000), statements after the matching case label are executed in sequence, regardless of the expression of subsequent case labels, until a break statement is encountered or end of the switch statement.
Propose a fix.

Solution: Add a `break` statement at the end of 100, 1000, 100000 cases.

6. (18 points) ConvertString
Assume that you are given a string in java program. Write a program that will print string back with masked digits (0-9) as asterisks (character '*'). All other characters such as letters or punctuation marks are not changed. For example, for string "Hello CS170 Students!", program should output "Hello CS*** Students!".

```java
public class ConvertString{
    public static void main(String [] args){
        String str;
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a string: ");
        str = input.nextLine();

        /*-------- YOUR CODE HERE -------->
        }
    }
}
```

Solution:
```java
int i = 0;
while(i < str.length()){
    if (str.charAt(i) >= '0' && str.charAt(i) <= '9'){
        System.out.print("*");
    }
    else{
        System.out.print(str.charAt(i));
    }
    i++;
}
System.out.println();
```
7. (18 points) Conversion table

Write a program using a while loop that prints conversion table from gallon to ounces and liters in range provided by user. 1 gallon equals 128 oz and 1 gallon equals 3.78541 liters. For input: 2 11 the output of program should look as follows:

Enter a range (e.g.: 1 10): 2 11
Gallon   Oz       Liter
2        256      7.57082
3        384      11.35623
4        512      15.14164
5        640      18.92705
6        768      22.71246
7        896      26.49787
8        1024     30.28328
9        1152     34.06869
10       1280     37.8541
11       1408     41.63951

Use a \t character to tabulate between strings. Example:

System.out.println("Column1" + "\t" + "Column2");
System.out.println(1 + "\t" + 2);

will give you:

Column1  Column2
1         2

Starter code is on the next page.
```java
public class ConversionTable{
    public static void main(String [] args){
        int start, end;
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a range: ");
        start = input.nextInt();
        end = input.nextInt();

        System.out.println("Gallon	Oz	Liter");
        /*-------- YOUR CODE HERE -------->

    }
}

Solution:
int oz;
double liter;

while(start <= end){
    oz = start * 128;
    liter = start * 3.78541;
    System.out.println(i + "\t" + oz + "\t" + liter);
    start++;
}
```
### ASCII Table

<table>
<thead>
<tr>
<th>0 NUL</th>
<th>1 SOH</th>
<th>2 STX</th>
<th>3 ETX</th>
<th>4 EOT</th>
<th>5 ENQ</th>
<th>6 ACK</th>
<th>7 BEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 BS</td>
<td>9 HT</td>
<td>10 LF</td>
<td>11 VT</td>
<td>12 FF</td>
<td>13 CR</td>
<td>14 SO</td>
<td>15 SI</td>
</tr>
<tr>
<td>16 DLE</td>
<td>17 DC1</td>
<td>18 DC2</td>
<td>19 DC3</td>
<td>20 DC4</td>
<td>21 NAK</td>
<td>22 SYN</td>
<td>23 ETB</td>
</tr>
<tr>
<td>24 CAN</td>
<td>25 SUB</td>
<td>26 WS</td>
<td>27 ESC</td>
<td>28 FS</td>
<td>29 GS</td>
<td>30 RS</td>
<td>31 US</td>
</tr>
<tr>
<td>32 SP</td>
<td>33 !</td>
<td>34 &quot;</td>
<td>35 #</td>
<td>36 $</td>
<td>37 %</td>
<td>38 &amp;</td>
<td>39 (</td>
</tr>
<tr>
<td>40 )</td>
<td>41 *</td>
<td>42 +</td>
<td>43 ,</td>
<td>44 .</td>
<td>45 /</td>
<td>46 0</td>
<td>47 1</td>
</tr>
<tr>
<td>48 2</td>
<td>49 3</td>
<td>50 4</td>
<td>51 5</td>
<td>52 6</td>
<td>53 7</td>
<td>54 8</td>
<td>55 9</td>
</tr>
<tr>
<td>56 0</td>
<td>57 1</td>
<td>58 2</td>
<td>59 3</td>
<td>60 4</td>
<td>61 5</td>
<td>62 6</td>
<td>63 7</td>
</tr>
<tr>
<td>64 @</td>
<td>65 A</td>
<td>66 B</td>
<td>67 C</td>
<td>68 D</td>
<td>69 E</td>
<td>70 F</td>
<td>71 G</td>
</tr>
<tr>
<td>72 H</td>
<td>73 I</td>
<td>74 J</td>
<td>75 K</td>
<td>76 L</td>
<td>77 M</td>
<td>78 N</td>
<td>79 O</td>
</tr>
<tr>
<td>80 P</td>
<td>81 Q</td>
<td>82 R</td>
<td>83 S</td>
<td>84 T</td>
<td>85 U</td>
<td>86 V</td>
<td>87 W</td>
</tr>
<tr>
<td>88 X</td>
<td>89 Y</td>
<td>90 Z</td>
<td>91 [</td>
<td>92 \</td>
<td>93 ]</td>
<td>94 ^</td>
<td>95 _</td>
</tr>
<tr>
<td>96 `</td>
<td>97 a</td>
<td>98 b</td>
<td>99 c</td>
<td>100 d</td>
<td>101 e</td>
<td>102 f</td>
<td>103 g</td>
</tr>
<tr>
<td>104 h</td>
<td>105 i</td>
<td>106 j</td>
<td>107 k</td>
<td>108 l</td>
<td>109 m</td>
<td>110 n</td>
<td>111 o</td>
</tr>
<tr>
<td>112 p</td>
<td>113 q</td>
<td>114 r</td>
<td>115 s</td>
<td>116 t</td>
<td>117 u</td>
<td>118 v</td>
<td>119 w</td>
</tr>
<tr>
<td>120 x</td>
<td>121 y</td>
<td>122 z</td>
<td>123 {</td>
<td>124</td>
<td>125 }</td>
<td>126 ~</td>
<td>127 DEL</td>
</tr>
</tbody>
</table>

Note that uppercase: $65 \leq x \leq 90$

Note that lowercase: $97 \leq x \leq 122$

Difference between A (65) and a (97) is 32!

Difference between Q (81) and q (113) is 32!