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 9 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>6</td>
<td>20</td>
<td>14</td>
<td>12</td>
<td>12</td>
<td>16</td>
<td>20</td>
<td>100</td>
</tr>
</tbody>
</table>

Score: ________________________________
1. (6 points) Definitions
For each of the following give a basic definition of the term. You do not need to give a formal definition.

- **break** keyword
  Break keyword is used to break the execution of the while loop or to break execution of switch statement and prevent fall through.

- **Unsafe conversion**
  Unsafe conversion is conversion when higher precision datatype is converted into lower precision datatype, for instance double (64 bit) to float (32 bit) or int (32 bit) to short (16 bit).

- **String**
  String is a datatype (not primitive) in Java and consists of characters. In Java, it is defined with double quotes "".

2. (20 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
1   Scanner scanner = new Scanner(System.in);
2   String s;
3   s = scanner.next();
4   // Now, user enters: 45.44
5   double d = s * 2;
6   System.out.println(d);
```

**Output or error:**
In line 5, s is String and is not a numerical datatype (it’s String), it has to be parsed first into the double.

- **Snippet 2**

```java
1   double d = 3.14;
2   int c = 12;
3   int f = c * d;
4   System.out.println(f);
```

**Output or error:**
In Line 3, there is loss of precision when double is assigned to int. It would have to contain explicit cast to int.

- **Snippet 3**

```java
1   double d = 4.0;
2
3   if (d == 2) {
```
4 \hspace{10pt} \texttt{System.out.println("First");}
5 \}
6 \textbf{else if} \left( d == 3 \right) \{
7 \hspace{10pt} \texttt{System.out.println("Second");}
8 \}
9 \textbf{else if} \left( d > 4 \right) \&\& \left( d < 10 \right) \{
10 \hspace{10pt} \texttt{System.out.println("Third");}
11 \}
12 \textbf{else} \{
13 \hspace{10pt} \texttt{System.out.println("Else");}
14 \}

\textbf{Output} or error:

Else

- \textbf{Snippet 4}

1 \hspace{10pt} \texttt{int d = 0;}
2 \texttt{while \left( d <= 20 \right) \{
3 \hspace{10pt} \texttt{System.out.println(d);}
4 \hspace{10pt} d = \left( d * 2 \right) + 1;
5 \}

\textbf{Output} or error:

0, 1, 3, 7, 15 (each after the new line)

- \textbf{Snippet 5}

1 \hspace{10pt} \texttt{double d = 3.14;}
2 \hspace{10pt} \texttt{int g = 10;}
3 \hspace{10pt} \texttt{int res = (((int) d * 4) \% g) + 1;}
4 \hspace{10pt} \texttt{System.out.println(res);}

\textbf{Output} or error:

3
3. (14 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 = 2.0, e = 4.0;
int i = 3, j = 15;
char c = 'x';
char d = 'd';
String str1 = "Hello world!"
```

<table>
<thead>
<tr>
<th>Java expression</th>
<th>Result</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>d + i</td>
<td>5.0</td>
<td>double</td>
</tr>
<tr>
<td>(double)(str1.length() + j)</td>
<td>27.0</td>
<td>double</td>
</tr>
<tr>
<td>++i + j - (3 * 4 % 2)</td>
<td>19</td>
<td>int</td>
</tr>
<tr>
<td>(char)('c' + 2*3)</td>
<td>'i'</td>
<td>char</td>
</tr>
<tr>
<td>str1 + &quot;45&quot; + '6'</td>
<td>Hello World!456</td>
<td>String</td>
</tr>
<tr>
<td>str1.substring(6, 11)</td>
<td>world</td>
<td>String</td>
</tr>
<tr>
<td>! (e &gt; d</td>
<td></td>
<td>i &gt; j)</td>
</tr>
<tr>
<td>(char)(str1.charAt(10) - 3)</td>
<td>'a'</td>
<td>char</td>
</tr>
</tbody>
</table>
4. (12 points) Problem 4
The Java code below runs on the machine. What will the output be if the user types:

```
Enter a number: 6
```

```java
int userInt;
int encoded_value;
Scanner input = new Scanner(System.in);
System.out.println("Enter a number: ");
userInt = input.nextInt();
switch(userInt){
   case 0:
   case 1:
   case 2:
       encoded_value = 1;
       break;
   case 3:
   case 4:
       encoded_value = 10;
       break;
   case 5:
       encoded_value = 100;
       break;
   case 6:
       encoded_value = 500;
   case 7:
       System.out.println("Case 7 has been processed");
   case 8:
       encoded_value = 1000;
       break;
   case 9:
       encoded_value = 10000;
       break;
   default:
       encoded_value = -1;
       break;
}
System.out.println("Between main two");
String message = "";
if (encoded_value <= 10000){
   if (encoded_value == -1){
       message = "Error in input ";
   }
   else{
       if (encoded_value <= 100){
           message = "Value is in <0; 100> range";
       } else if (encoded_value == 1000 || encoded_value == 1001){
           message = "Value is in the <1000; 1001> range";
       } else{
           message = "Value greater than 1001";
       }
   }
} else{
   message = "Error in decoding your value";
}
System.out.println("End of choices");
System.out.println("Message is: " + message);
```
Output:
Case 7 has been processed
Between main two
End of choices
Message is: Value is in the <1000; 1001> range

5. (12 points) Problem 5
Consider the code below with logical errors that should calculate the sum of even numbers from the range provided by user. For instance, for numbers 4 and 12, it should sum up numbers: 4, 6, 8, 10, 12 and print the sum, which is 40.

Enter start number: 4
Enter end number: 12
Sum of even numbers is: 40

```java
import java.util.Scanner;

public class SumEvenAns{
    public static void main(String [] args){
        int start_number, end_number;
        int sum = 0;
        Scanner input = new Scanner(System.in);
        System.out.print("Enter start number: ");
        start_number = input.nextInt();
        System.out.print("Enter end number: ");
        end_number = input.nextInt();
        while(start_number <= end_number){
            if (start_number % 2 == 0){
                sum = sum + start_number;
            }
            start_number++;
        }
        System.out.println("Sum of even numbers is: " + sum);
    }
}
```

- What does the code output for input: 5 and 9 (start/end number)?
  35
- Explain the logical error in the code.
  While loop contains a logical error, because it is summing all numbers, even and odd ones.
- Propose a fix.
  Additional if should be placed (see the code).
6. (16 points) CalculateCharacter
Assume that you are given a string and a character in Java. Write a program that will count the number of occurrences of the character in the string. For instance, if user enters "Hello world!" as a string and character 'l', the program should output: *There is 3 characters 'l' in string Hello World!*

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

        System.out.print("Enter a string: ");
        str = input.next();
        System.out.print("Enter a character: ");
        c = input.next().charAt(0);

        /*-------- YOUR CODE HERE -------->
        int i = 0;
        int count = 0;

        while (i < str.length()){
            if (str.charAt(i) == c){
                count++;
            }
            i++;
        }

        System.out.println("There is " + counter + 
        " characters '" + c + '" in string " + str);
    }
}
```
7. (20 points) DivisorsAverage

Write a program using a while loop that finds an average of all divisors of a number entered by user. For instance, for input 12, all the divisors are: 1, 2, 3, 4, 6, 12 and they sum up to 28. Then, 28/6 (number of divisors) will give a value of 4.6667, so the output of program should look as follows:

Enter a number: 12
Average of all divisors: 4.6667

```java
public class DivisorsAverage{
    public static void main(String [] args){
        int start, end;
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a number: ");
        number = scanner.nextInt();

        /*-------- YOUR CODE HERE -------->

        int i = 1, divisors = 0, sum = 0;

        while (i <= number){
            if (number % i == 0){
                sum += i;
                divisors ++;
            }
        }

        i++;
    }

    double average = (double)sum/divisors
    System.out.println("Average of all divisors: " + average);
}
}
```
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 NL</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 ECH</td>
<td>26 SUB</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 .</td>
<td>47 /</td>
</tr>
<tr>
<td>48 0</td>
<td>49 1</td>
<td>50 2</td>
<td>51 3</td>
<td>52 4</td>
<td>53 5</td>
<td>54 6</td>
<td>55 7</td>
</tr>
<tr>
<td>56 8</td>
<td>57 9</td>
<td>58 :</td>
<td>59 ;</td>
<td>60 &lt;</td>
<td>61 =</td>
<td>62 &gt;</td>
<td>63 ?</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 <= x <= 90
Difference between A (65) and a (97) is 32!
Note that lowercase 97 <= x <= 122
Difference between Q (81) and q (113) is 32!