• 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 9 questions on 11 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>9</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points:</td>
<td>9</td>
<td>14</td>
<td>10</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>10</td>
<td>76</td>
</tr>
<tr>
<td>Score:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. (9 points) For each of the following give a basic definition of the term. You do not need to give a formal definition. Then, give an example of the error in Java code. (In other words, write a snippet of code which demonstrates the type of error.)

(a) syntax error

**Solution:** An error the compiler will catch that violates the rules of the language.

Example: missing semicolon:
```java
System.out.println("Hi")
```

(b) runtime error

**Solution:** An error that occurs as the program runs and prevents the program from completing.

Example: index out of bounds
```java
String s = "Hello"; char c = s.charAt(5);
```

(c) logic (or logical) error

**Solution:** a programmer error. The program is syntactically correct, but it fails to arrive at the correct answer through an error in programmer understanding or logic.

Example: confusing % and \ 
```java
if(num / 2 == 0) to identify even numbers
```
2. (14 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 = "Hello", s2 = "World", s3 = "17";
char c1 = ‘_’, c2 = ‘!’;
int i1 = 4, i2 = 5;
double d1 = 2.0, d2 = 7.0;
```

<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>i1 + i2 * i1</td>
<td>24</td>
<td>int</td>
</tr>
<tr>
<td>i1 + i2 / i1</td>
<td>5</td>
<td>int</td>
</tr>
<tr>
<td>i1 % i2</td>
<td>4</td>
<td>int</td>
</tr>
<tr>
<td>d1 / i1 + i2</td>
<td>5.5</td>
<td>double</td>
</tr>
<tr>
<td>d1 != i1/2</td>
<td>false</td>
<td>boolean</td>
</tr>
<tr>
<td>d2 * i1 &lt;= 29 - i2%2</td>
<td>true</td>
<td>boolean</td>
</tr>
<tr>
<td>(int)s3</td>
<td>error</td>
<td></td>
</tr>
<tr>
<td>(int)d2/d7</td>
<td>3.5</td>
<td>double</td>
</tr>
<tr>
<td>s1 + c1 + s2 + c2</td>
<td>Hello_World!</td>
<td>String</td>
</tr>
<tr>
<td>s1 + i1 + d2</td>
<td>Hello47.0</td>
<td>String</td>
</tr>
<tr>
<td>d2 + i1 + s1</td>
<td>11.0Hello</td>
<td>String</td>
</tr>
<tr>
<td>s1.substring(1,4)</td>
<td>ell</td>
<td>String</td>
</tr>
<tr>
<td>Integer.parseInt(s3+i1)</td>
<td>174</td>
<td>int</td>
</tr>
<tr>
<td>i2 + Integer.toString(i2)</td>
<td>55</td>
<td>String</td>
</tr>
</tbody>
</table>
3. (10 points) Assume the statements below are part of a Java program which compiles and runs. What is the output if the user types 20?

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

System.out.println("one");
if (num <= 40) {
    System.out.println("hi");
} else if (num <= 20) {
    System.out.println("cry");
} else if (num <= 10) {
    System.out.println("bye");
} else {
    System.out.println("sigh");
}

System.out.println("two");
if (num > 30) {
    System.out.println("hi");
} else {
    System.out.println("cry");
} if (num > 20) {
    System.out.println("bye");
} else {
    System.out.println("sigh");
}

System.out.println("three");
switch(num):
    case 20:
        System.out.println("hi");
    case 15:
        System.out.println("cry");
    case 10:
        System.out.println("bye");
        break;
    case 5:
        default:
            System.out.println("sigh");

System.out.println("four");
if (num % 10 == 0) {
    System.out.println("hi");
}
if (num % 20 == 0) {
    System.out.println("cry");
}
if (num % 4 == 5) {
    System.out.println("bye");
} else {
    System.out.println("try");
}
```
4. (6 points) For each of the following, exactly how many times does each loop execute, assuming there are no statements that modify the integer variable \( i \) inside the body of the loop?

(a) \( \text{for}(i = 0; i < 10; i++) \{\ldots\} \)

Solution: 10

(b) \( \text{for}(i = -10; i <= 10; i++) \{\ldots\} \)

Solution: 21

(c) \( \text{for}(i = 10; i >= -10; i = i-2) \{\ldots\} \)

Solution: 11

5. (8 points) When executed in a program, which of the following code snippets have the same output? You may assume all snippets are executed with the same input.

(a) \( \text{for} \ (\text{int} \ i = 0; \ i < n; \ i++) \{ \)
   \hspace{1em} \( \text{System.out.println}(i+1); \)
   \hspace{1em} \} \)

(b) \( \text{int} \ i = 1; \)
   \hspace{1em} \( \text{while} \ (i < n) \{ \)
   \hspace{2em} \( \text{System.out.println}(i); \)
   \hspace{2em} \( i++; \)
   \hspace{1em} \} \)
(c) int i = 1;
    while (i < n) {
        System.out.println(i+1);
        i++;
    }

(d) int i = 0;
    while (i < n) {
        i = i + 1;
        System.out.println(i);
    }

**Solution:** a and d give identical output
Scoring: -2 for each incorrect inclusion or omission
6. Consider the following program (with the lines numbered for convenient reference).

```java
1: public class Bugs {
2:    public static void main(String args[]) {
3:        int letter_count = 0;
4:        String check_word = "Debugging";
5:        char single_letter = '"';
6:        for(int i = 0; i < check_word.length(); i++) {
7:            single_letter = check_word.charAt(1);
8:            if (single_letter == 'g') {
9:                letter_count++;
10:            }
11:        }
12:        System.out.println("There are " + letter_count + "gs.");
13:    }
14: }
```

The intent of the program is to count the number of lowercase g’s in a given String. However, when the given String is “Debugging” as in the program above, it prints out “There are 0 gs” which is obviously incorrect.

(a) (2 points) Explain what the error is in the program and why it occurs.

**Solution:** The error is on line 8. `check_word.charAt(1)` will always be the letter 'e'. When compared to 'g', the comparison is always false and `letter_count` never increments.

(b) (4 points) What line(s) of code would you change to eliminate this error? Change the line(s) to make the program function correctly, assuming that `limit` is any valid positive integer.

**Solution:** Easiest fix is to change line 8 to:

```
    single_letter = check_word.charAt(i);
```
7. (7 points) The program below accepts 3 integers from the user. You should add code to figure out if the user entered the numbers in **increasing** order, unless all three numbers are equal. Your program should print out either “Numbers increase in order”, “Numbers do not increase”, or “All numbers are equal”. You may assume the user only enters integer values in the terminal window.

```java
import java.util.Scanner;

public class InOrder {
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);

        int a = in.nextInt();
        int b = in.nextInt();
        int c = in.nextInt();

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

    }
}
```

**Solution:**

```java
if (a == b && b == c)
    System.out.println(‘’All numbers are equal’’);
else if (a < b && b < c)
    System.out.println(‘’Numbers increase in order’’)
else
    System.out.println(‘’Numbers do not increase’’);
```

**Scoring:**

+2 identifies (and prints) all numbers equal correctly
+2 identifies (and prints) numbers increasing correctly
+2 identifies (and prints) numbers not increasing correctly
+1 correct use/structure of if statement(s)
8. (6 points) Complete the program below. The program prompts the user for a number of inches. You should print out the conversion to feet and inches. (There are 12 inches in 1 foot.) You may assume the user always enters a valid, positive integer. Some examples of the output are below:

Examples:

```java
>>> java FeetIn
Enter the number of inches: 15
15 inches is 1 feet 3 inches.
>>> java FeetIn
Enter the number of inches: 28
28 inches is 2 feet 4 inches.
```

```java
import java.util.Scanner;

public class FeetIn {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        System.out.print("Enter the number of inches: ");
        int inches = in.nextInt();
        /*----------- Your code here -------------*/
        int feet = inches / 12;
        int rem_inches = inches % 12;
        System.out.println(inches + " inches is " + feet + " feet " + rem_inches + " inches");
    }
}
```

Solution:

```java
int feet = inches / 12;
int rem_inches = inches % 12;
System.out.println(inches + " inches is " + feet + " feet " + rem_inches + " inches");
```

Scoring:
+2 Calculates number of feet correctly
+2 Calculates number of inches correctly
+2 Displays info correctly
9. You want to write a program that “doubles” a String. This means that each character in the string is repeated twice, letter by letter. The program has been started for you below. Some examples are also given below.

Examples:

```java
>>> java DoubleStr
Enter a word: This
TT
TThh
TThhiiss

>>> java Double Str
Enter a word: hi
hh
hhii
TThhiiss
```

```java
import java.util.Scanner;

public static DoubleStr {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        System.out.println("Enter a word: ");
        String s = in.next();

        /*------ Your code for part a OR part b here -------*/

        } }
}
```

(a) (5 points) Fill in the code which “doubles” the string. You **must** use a `while` loop for this part.

**Solution:**

```java
int i = 0;
String result = "";
while (i < s.length()) {
    result += s.charAt(i);
    result += s.charAt(i);
    System.out.println(result);
    i++;
}
```

On this problem, you have to be careful “adding” characters. This will result in an integer (rather than the intended concatenation) if you aren’t careful.
Scoring:
+1 while loop used (correctly)
+1 uses “sybmol” of character rather than integer value
+2 String printed out incrementally (see problem examples)
+1 Last iteration of String doubled

(b) (5 points) Fill in the code which “doubles” the string again, but this time you must use a for loop.

**Solution:** A different (but equally correct) solution than above:

```java
String result = "";
for(int i = 0; i < s.length(); i++)
    String s = s.substring(i,i+1);
    s = s+s;
    result = result + s;
System.out.println(result);
```

Scoring:
+1 for loop used (correctly)
+1 uses “sybmol” of character rather than integer value
+2 String printed out incrementally (see problem examples)
+1 Last iteration of String doubled