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 10 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>5</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>6</td>
<td>9</td>
<td>4</td>
<td>50</td>
</tr>
</tbody>
</table>

Score: 

1
1. (5 points) Define each of the 5 terms below. You do not need to give a formal definition, just a good description. You may include an example if it is helpful to your answer. Be brief – give a maximum of two sentences.

(a) variable

**Solution:** Variables are used to store data in a program. A variable represents a value that a programmer wants to store and access and/or change later in a program.

(b) compiler

**Solution:** A compiler is software which translates a high level language to machine language.

(c) operator

**Solution:** An operator is a symbol which causes some sort of action (eg. assignment) or computation to occur.

(d) operator associativity

**Solution:** If operators with the same precedence are next to each other, their associativity determines the order of evaluation.

(e) syntax error

**Solution:** A syntax error violates the grammatical rules of the programming language being used. Syntax errors are caught by the compiler.

(f) logic (or logical) error

**Solution:** A logical error is an error in a program that compiles and runs, but does not achieve the intended goal of the program.

```java
for (int x = 4; x > 0; x++) {
    ...
}
```

*Rubric:* +1 point for each correct answer; +0.5 for partially correct answers. (0 points for a definition primarily depending on a variant of the word itself.)
2. (9 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, and write the type of error (syntax, runtime, or logic). The first row has been done for you.

```java
String s1 = "Mardi", s2 = "Gras", s3 = "2015";
char c1 = '1', c2 = 'A';
int i1 = 3, i2 = 8;
double d1 = 1.0, d2 = 3.5, d3 = 0.5;
```

<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>s2.charAt(i1+1)</td>
<td>Error</td>
<td>runtime</td>
</tr>
<tr>
<td>s1.charAt(i2-i1)</td>
<td>Error</td>
<td>runtime</td>
</tr>
<tr>
<td>i1 / i2</td>
<td>0</td>
<td>int</td>
</tr>
<tr>
<td>d1 ** 10 &gt;= i1</td>
<td>Error</td>
<td>syntax</td>
</tr>
<tr>
<td>s3 + i1 * (int)d2</td>
<td>&quot;20159&quot;</td>
<td>String</td>
</tr>
<tr>
<td>d2 + i2 + s3</td>
<td>&quot;11.52015&quot;</td>
<td>String</td>
</tr>
<tr>
<td>(char)(c2 + 4)</td>
<td>'E'</td>
<td>char</td>
</tr>
<tr>
<td>d2 &lt; i1+i2</td>
<td></td>
<td>d2 == i1+i3*3</td>
</tr>
<tr>
<td>s2 + c1 + c2</td>
<td>&quot;Gras1A&quot;</td>
<td>String</td>
</tr>
<tr>
<td>!(d1 + i1 + d3 &gt;= d2)</td>
<td>false</td>
<td>boolean</td>
</tr>
<tr>
<td>i2 &gt; i1</td>
<td></td>
<td>!((d2 &gt; d1)</td>
</tr>
<tr>
<td>s1 + &quot;&quot;&quot; + s3 + &quot;&quot;&quot;&quot;</td>
<td>&quot;Mardi&quot;2015&quot;&quot;</td>
<td>String</td>
</tr>
</tbody>
</table>

Notes:

- The parenthesis operator has the highest precedence of all operators.
- The logical not operator and the casting operator have higher precedence than arithmetic operators, relational/comparison operators, and the other logical operators.
- Arithmetic operators have higher precedence than relational/comparison or logical operators.
- Relational/comparison operators have higher precedence than logical operators.
- Assignment operators have the lowest precedence of all operators

*Rubric:* 0.5 points for each correct answer (each cell of the table).
3. (8 points) Assume the statements below are part of a Java program which compiles and runs. What is the output if the user enters 10 and 30 in that order?

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

if(a <= 40 && b >= 40) {
    System.out.println("blue");
} else if (a >= 40 || b >= 40) {
    System.out.println("green");
} else if (a <= 40 || b >= 40) {
    System.out.println("red");
} else {
    System.out.println("yellow");
}

if (!(a <= 30) && b >= 20) {
    System.out.println("pineapple");
} else {
    System.out.println("pear");
} if (a >= 30 && b >= 20) {
    System.out.println("banana");
} else {
    System.out.println("cranberry");
}

switch(a * b/2) {
    case 200:
        System.out.println("head");
    case 150:
        System.out.println("shoulders");
    case 100:
        System.out.println("knees");
        break;
    default:
        System.out.println("toes");
}

if ((a-b) == 20 || (b-a) == 20) {
    System.out.println("dog");
} if ((b-a) >= a ) {
    System.out.println("cat");
} if (a == 10 && b != 10) {
    System.out.println("fish");
} else {
    System.out.println("iguana");
}
```
Solution:
red
pear
cranberry
shoulders
knees
dog
cat
fish

Rubric: 1 point for each correctly recorded output.
4. For each of the code fragments below, give the output. If the code results in an infinite loop, write the first few outputs, and then indicate that it is an infinite loop.

(a) (2 points)
```java
for(int i = 3; i <= 10; i++) {
    System.out.println(i);
    i += 2;
}
```

Solution:
3
6
9

(b) (2 points)
```java
int i = 0;
while (i < 10) {
    System.out.println(i);
    if (i == 6) {
        continue;
    } else {
        i = i+2;
    }
    i++;
}
```

Solution:
0
3
6
infinite loop

(c) (2 points)
```java
int x = 10;
while(x >= 0) {
    System.out.println("x: "+ x);
    if (x % 4 == 0) {
        x += 3;
        break;
    } else {
        x--;
        continue;
    }
}
System.out.println("Final value: "+ x);
```

Solution:
x: 10
x: 9
x: 8
Final value: 11
(d) (3 points)
    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 % b == 0);
                a += b;
            case 5:
                System.out.println(a - b);
                break;
        }
        if (a % 7 == 0) {
            System.out.println(a);
            break;
        }
        a++;
    }

    Solution:
    1
    true
    4
    7
5. (6 points) Find the errors in the following program “Midterm1.java”

you can rewrite the statement, insert a statement, or delete a statement. Use the line number on the
left to indicate the location of the code you want to change. To insert a statement, you need to provide
two adjacent line numbers between which you wish to insert your statement.

Description of the program: read a sequence of positive integers in from the terminal. When the user
types 0, the program will print out the largest number entered and exit.

1. public class exam {
2.     public static void main(String[] args) {
3.         Scanner in = new Scanner(System.in);
4.         System.out.println("Enter a sequence of positive integers ending with a zero");
5.         number = in.nextInt();
6.         largest = number;
7.         while(number > 0) {
8.             if (largest < number)
9.                 largest = number;
10.         }
11.         System.out.println("The largest number entered was: " + largest);
12.     }
13. }

Solution:

1. public class Midterm1 { // change exam to Midterm1

5.     int number = in.nextInt(); // Declare number to be an integer
6.     int largest = number; // Declare number to be an integer
7.     while(number > 0) { // Remove equal sign to correct logic error
8.         if (largest < number) // Remove the semi-colon
9.             largest = number; // Assign largest to the value of number

insert (between lines 9 and line 10) a statement to read in the next integer:

         number = in.nextInt();

6. (9 points) Complete the program below. The program should read in an integer entered by the user. If the integer is negative, the program should print out the odd digits in the integer separated by a space. (If there are no odd digits, the program should print "none".) If the integer is non-negative, the program should print out the even digits in the integer separated by a space. (If there are no even digits, the program should print "none".) Examples are below:

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>170002</td>
<td>0 0 0 2</td>
</tr>
<tr>
<td>-170002</td>
<td>1 7</td>
</tr>
<tr>
<td>2014</td>
<td>2 0 4</td>
</tr>
<tr>
<td>-40</td>
<td>none</td>
</tr>
</tbody>
</table>

Solution:

```java
import java.util.Scanner;

public class EvenOddDigits {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter an integer: ");

        int number = in.nextInt();
        String desireddigits = "";

        if (number < 0) {
            number = -number; // convert number to its absolute value
            while (number > 0) {
                if (number % 2 != 0)
                    desireddigits = (number % 10) + " " + desireddigits;
                number = number/10;
            }
        }
        else {
            while (number > 0) {
                if (number % 2 == 0)
                    desireddigits = (number % 10) + " " + desireddigits;
                number = number/10;
            }
        }
        System.out.println(desireddigits);
    }
}
```
7. (4 points) Determine the purpose of the following program. State the output if a user enters the strings “Mississippi” and ’is”, in that order. State the output if a user enters the strings “mathematics” and ’mat”, in that order.

```java
import java.util.Scanner;

public class matchtest {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String string1, string2;
        int position;
        boolean myboolean = false;

        System.out.print("Enter a string: ");
        string1 = in.nextLine();
        System.out.print("Enter another string: ");
        string2 = in.next();

        if (string1.length() >= string2.length()){
            for (position = 0; position <= string1.length()-string2.length(); position++) {
                myboolean = true;
                for (int j = 0; j < string2.length(); j++){
                    if (string1.charAt(position+j) != string2.charAt(j) ) {
                        myboolean = false;
                        break;
                    }
                }
                if (myboolean)
                    System.out.println("position: " + position);
            }
        }
    }
}
```

Solution: The program searches for substrings in the first string entered by the user that match the string in the second string entered by the user. If a match is found the program outputs the position of the first character in the matching substring.

Output if a user enters the strings “Mississippi” and ’is”, in that order:

position: 1
position: 4

Output if a user enters the strings “mathematics” and ’mat”, in that order:

position: 0
position: 5