Name (as on OPUS): ____________________________________________________________

Section: ___________________ Seat Assignment: ________________________________

• 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.

• ANSWERS: Answers for problems 1-33 MUST be on your Scantron sheet. Answers written on your exam booklet will not be considered. You should take care to mark your answers neatly. If the machine cannot read an answer, it is considered incorrect. Answers must be recorded on your Scantron sheet before time is called.

• TIME: This exam has 2 parts on 12 pages including the title page. Please check to make sure all pages are included. You will have 65 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>Multiple Choice</th>
<th>Code writing</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points:</td>
<td>51</td>
<td>14</td>
<td>65</td>
</tr>
<tr>
<td>Score:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Part 1: Multiple Choice
For each question, clearly indicate the best answer on your Scantron sheet. Only answers marked correctly on the Scantron sheet will be scored; the exam booklet will not be considered. Scantrons must be completed before time is called.

For questions 1-10, evaluate each expression given the variables and values below. Choose the correct letter for the resulting value from the chart below. Results in the chart indicate datatype: 4.0 is different then 4 is different then "4.0"! The first is a double, the second is an int, and the last is a String. If the expression cannot be evaluated or is not proper Java syntax, you may simply choose “error” (answer Z) for the value.

```java
String s1 = "Fall", s2 = "15";
int i1 = 7, i2 = 2, i3 = 4;
double d1 = 6.6, d2 = 3.8;
```

(1) (1 point) (int) d1 + d2

Solution: B

(2) (1 point) s1 + i1 * i2

Solution: I

(3) (1 point) s1 + i1 + 1

Solution: G

(4) (1 point) s2 * 2

Solution: Z

(5) (1 point) i1 / i2 * i3

Solution: W

(6) (1 point) i1 * i3 / i1

Solution: R

(7) (1 point) i3 % i1

Solution: R

(8) (1 point) d1 < d2 || i2 < i3

Solution: N

(9) (1 point) i1 > i2 && i3 == i2
Solution: O

(10) (1 point) \( i_1 > i_3 > i_2 \)

Solution: Z

Possible Answers: (letter to bubble in is above value)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9.6</td>
<td>9.8</td>
<td>10.0</td>
<td>10.4</td>
<td>14.0</td>
</tr>
<tr>
<td>F</td>
<td>“Fall72”</td>
<td>“Fall71”</td>
<td>“Fall8”</td>
<td>“Fall14”</td>
<td>“Fall15”</td>
</tr>
<tr>
<td>K</td>
<td>“true”</td>
<td>9</td>
<td>“false”</td>
<td>4</td>
<td>“30”</td>
</tr>
<tr>
<td>P</td>
<td>“true”</td>
<td>0</td>
<td>“false”</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>U</td>
<td>“true”</td>
<td>9</td>
<td>“false”</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>Z</td>
<td>“true”</td>
<td>30</td>
<td>“false”</td>
<td>14</td>
<td>30</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.
Questions 11-14 deal with commands in Linux and reference the following directory hierarchy. Assume that you can create any file/directory in any directory.

(11) (2 points) You want to create a new directory named `case` inside the `bin` directory. If you are currently in the `cs170` directory, which of the following commands will accomplish this?

A. `touch /bin/case`
B. `touch bin/case`
C. `mkdir /bin/case`
D. `mkdir bin/case`
E. `/bin/case`

(12) (2 points) In Linux, which command will take you to your home directory?

A. `cd home`
B. `cd`
C. `cd ..`
D. `cd /`
E. `cd ../../`

(13) (2 points) Which Linux command displays the contents of a directory?

A. `dir`
B. `show`
C. `list`
D. `ls`
E. `touch`

(14) (2 points) You are in the `home` directory. Which of the following is an absolute path to the `Exam.java` file?

A. `/home/ Elijah/cs170/Exam.java`
B. `/home/ Elijah/cs170/
C. `elijah/cs170/Exam.java`
D. `/elijah/cs170/Exam.java`
E. `home/ Elijah/cs170/Exam.java`
Questions 15-22 refer to the following program:

```java
public class Scope {
    public static int q = 3;

    public static void method1(int b) {
        int a = 6;
        System.out.println(a); /*1*/
    }

    public static int method2(int q) {
        System.out.println(q); /*2*/
        return q - Scope.q;
    }

    public static void main(String[] args) {
        int a = 2;
        method1(a);
        System.out.println(a); /*3*/
        q = method2(4);
        System.out.println(q); /*4*/
        if (q < 10) {
            int q = 12;
            System.out.println(q); /*5*/
        }
        System.out.println(q); /*6*/
    }
}
```

(15) (1 point) What is displayed when the print statement at /*1*/ executes?
A. variable reference will cause an error
B. 2
C. 3
D. 4
E. 6

(16) (1 point) What is displayed when the print statement at /*2*/ executes?
A. variable cannot be used at this point; variable out of scope
B. 2
C. 3
D. 4
E. 6

(17) (1 point) What is displayed when the print statement at /*3*/ executes?
A. variable cannot be used at this point; variable out of scope
B. 2
C. 3
D. 4
E. 6
(18) (1 point) What is displayed when the print statement at /*4*/ executes?
   A. variable cannot be used at this point; variable out of scope
   B. 1
   C. -1
   D. 3
   E. 4

(19) (1 point) What is displayed when the print statement at /*5*/ executes?
   A. variable cannot be used at this point; variable out of scope
   B. 1
   C. 3
   D. 4
   E. 12

(20) (1 point) What is displayed when the print statement at /*6*/ executes?
   A. variable cannot be used at this point; variable out of scope
   B. 1
   C. 3
   D. 4
   E. 12

(21) (1 point) How many parameter variables appear in the code above?
   A. 0
   B. 1
   C. 2
   D. 3
   E. 4

(22) (1 point) How many local variables appear in the main method?
   A. 0
   B. 1
   C. 2
   D. 3
   E. 4
Questions 23-26 reference the following code snippet:

```java
int val = //some initial value
if( val > 4 ) {
    System.out.println( "Test A" );
} else if( val > 9 ) {
    System.out.println( "Test B" );
} else if ( val < 0 ) {
    System.out.println( "Test C" );
} else {
    System.out.println( "Test D" );
}
```

(23) (2 points) Which of the following initial values of `val` will result in “Test A” being printed?
A. -5
B. 0
C. 3
D. 8
E. There is no initial value which will result in “Test A” being printed

(24) (2 points) Which of the following initial values of `val` will result in “Test B” being printed:
A. -5
B. 3
C. 8
D. 12
E. There is no initial value which will result in “Test B” being printed

(25) (2 points) Which initial value of `val` will result in “Test C” being printed:
A. -5
B. 0
C. 3
D. 8
E. There is no initial value which will result in “Test C” being printed

(26) (2 points) Which initial value of `val` will result in “Test D” being printed:
A. -5
B. 0
C. 8
D. 12
E. There is no initial value which will result in “Test D” being printed
Questions 27-28 reference the following code snippet which contains line numbers:

```java
1 String e = "2.718";
2 double pi = 3.14;
3 String pie = pi + e;
4 System.out.println( (int)(pi) );
5 System.out.println( (int)(e) );
6 System.out.println( Double.parseDouble(pie) );
7 System.out.println( pie );
```

(27) (2 points) Which line number will cause a **compilation** (eg syntax) error?
A. line 3  
B. line 4  
C. line 5  
D. line 6  
E. line 7  

(28) (2 points) Which line number will cause a **runtime** error?
A. line 3  
B. line 4  
C. line 5  
D. line 6  
E. line 7  

(29) (1 point) You are given the following program:

```java
import java.util.Scanner;
public class SomeClass {
    //code for the class here
}
```

What is the name of the java file containing this program?
A. Scanner.java  
B. SomeClass.class  
C. SomeClass  
D. SomeClass.java  
E. Any file name with a .java extension will do

(30) (3 points) What will the following code print to the screen?

```java
int x = 15;
System.out.print("x \n");
System.out.println(x);
```

A. x \n15  
B. x \n15  
C. x  
15  
D. x 15  
E. The code will not run; The code contains an error.
31) (3 points) Consider this class example:

```java
public class MyPoint {
    public static void switchCoords(int x, int y) {
        int temp;
        temp = x;
        x = y;
        y = temp;
        System.out.print("(" + x + ", " + y + ")");
    }

    public static void main(String[] args) {
        int x, y;
        x = 5; y = 3;
        System.out.print("(" + x + ", " + y + ")");
        switchCoords(x, y);
        System.out.println("(" + x + ", " + y + ")");
    }
}
```

What is printed to when the program is executed?
A. (5, 3) (3, 5) (5, 3)  
B. (5, 3) (5, 3) (5, 3)  
C. (5, 3) (3, 5) (3, 5)  
D. (5,3) (3, 3) (3, 3)  
E. (5,3) (3, 3) (5, 3)  

32) (3 points) Consider this expression in which high and low are ints:

```
((int)(Math.random() * (high - low + 1))) + low
```

Which statement describes the range of numbers which can result from evaluating the expression?
A. all integers between the values low and high, inclusive  
B. all integers between 0 and high, inclusive  
C. all integers between low (inclusive) and high (exclusive)  
D. all integers between 0 (inclusive) and high (exclusive)  
E. all integers between low and high but not including either one  

33) (3 points) Each statement below is intended to print “number ok” if number is any value less than 0 or greater than 100. Nothing should be printed if the value is in the range 0...100, inclusive. Which implementation is correct?
A. `if (number <= 0 || number >= 100) {
    System.out.println("number ok");
}
`  
B. `if (! (number <= 0 && number >= 100)) {
    System.out.println("number ok");
}
`  
C. `if (number < 0 || number > 100) {
    System.out.println("number ok");
}
`  
D. `if (number >= 0 && number <= 100) {
    System.out.println("number ok");
}
`  
E. `if (0 > number && 100 > number) {
    System.out.println("number ok");
}
`
Part 2: Code writing

Write valid Java code to solve the following problems. Answers to this section should be written in your exam booklet.

(a) (1 point) Write your user id here (what you login with; not your ID number):

(b) (6 points) Write a method named `fizzBuzz` that takes a single integer parameter. Your method should always return the same value as your input parameter value. In addition, your method should print out the value of the input parameter except for the cases below, when your method should print a word:

- If the number is a multiple of three, you should print “Fizz” instead of a number.
- If the number is a multiple of five, you should print “Buzz” instead of a number.
- If the number is a multiple of both three and five, you should print out “FizzBuzz” instead of a number.

You may assume the input argument value is greater than or equal to 1. Examples:

- `fizzBuzz(15)` prints `FizzBuzz` and returns `15`
- `fizzBuzz(5)` prints `Buzz` and returns `5`
- `fizzBuzz(2)` prints `2` and returns `2`

Solution:

```java
public static int fizzBuzz(int n) {
    if (n % 3 == 0 && n % 5 == 0) {
        System.out.println("FizzBuzz");
    } else if (n % 3 == 0) {
        System.out.println("Fizz");
    } else if (n % 5 == 0) {
        System.out.println("Buzz");
    } else {
        System.out.println(n);
    }
    return n;
}
```

Scoring:

+1 correct function header
+1 prints FizzBuzz for multiples of 3 AND 5 (and no others!)
+1 prints Buzz for multiples of 5 (and no others!)
+1 prints Fizz for multiples of 3 (and no others!)
+1 prints out num for other numbers which are not multiples of 3 and 5.
-1 minor syntax errors
-2 major syntax errors

Common errors:
Most common was an ordering problem. For example, changing the code to something like:

```java
    if (n % 3 == 0) {
        System.out.println("Fizz");
    } else if (n % 5 == 0) {
        System.out.println("Buzz");
    } else if (n % 3 == 0 && n % 5 == 0) {
```
System.out.println("FizzBuzz");
}

Means that multiples of 3 AND 5 (eg. 15) will match the first condition since 15%3 == 0.

Another problem was trying to use the \ operator instead of the % operator.

Some students also confused return and println statements.

Some students put a return statement either 1) as the first line of the method or 2) before the println stmt inside the if stmt. Either of these will cause the method to terminate prematurely as return statements will end the method execution immediately.
(c) (7 points) Write a method named `unique` which takes 3 integer values as parameters. The method should return the number of unique numbers among the three. Examples:

- `unique(3, 4, 5)` returns 3 because all the numbers are different.
- `unique(4, 4, 5)` returns 2 because there are only 2 different numbers.

Solution:

```java
public static int unique(int a, int b, int c) {
    if (a == b && b == c) {
        return 1;
    } else if (a == b || b == c || a == c) {
        return 2;
    } else {
        return 3;
    }
}
```

Scoring:

- +1 method header correct
- +2 method returns 1 if all three values are the same (and only if all 3 are the same)
- +2 method returns 2 if two are the same (but not all 3)
- +2 method returns 1 if all are different.
- -1 minor syntax errors
- -2 major syntax errors

Common errors:
The most common error was not testing if `a==c` as in the case of `unique(2, 1, 2)`.

Also common was an ordering problem. Some students did something like:

```java
    if (a == b || b == c || a == c) {
        return 2;
    } else if (a == b && b == c) {
        return 1;
    } ...
```

In this case, the first condition will match if all 3 parameters have the same value (eg. `unique(2, 2, 2)`) because `a==b`. This will cause the incorrect value to be returned. Ordering issues can take other forms as well, but this was the most common one.