CS 170 (000) – Introduction to Computer Science
Midterm Exam #1
Mon, Feb 25 (10:40am – 11:30am)

Student name: ____________________________    Student ID: __________________________

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
• Write neatly and clearly indicate your answers. What I cannot read, I will assume to be incorrect.

Time: There are 5 problems and you will have 50 minutes to complete this exam. Please check to make sure no page is missing.

Honor code: Like all work for this class, the Emory Honor Code applies. Initial here to indicate that you followed the Honor Code and this work is your own: __________

Self-evaluation: After you complete the exam, please estimate your performance. How many points do you think you will get:

_____________________________________

_____________________________________
Problem 1 – Introduction to computers, programming and Java (10 pts)
Please write short (no more than one or two sentences) answers to the following questions:

a) What is a computer program?
sequence of instructions written to perform a specified task

b) What kind of data does the RAM store? What can this data represent?
RAM consists of binary switches, which store binary values 0 or 1 (bits). Bits group together to represent numbers, etc. In RAM you can store anything that can be encoded in binary format.

c) What is the Program Counter?
A special purpose CPU register that stores the address of the next instruction.

d) List the types of CPU instructions:
Arithmetic and logic; memory transfer; branch instructions

e) How strings are stored in a computer memory?
As a sequence of characters, each character is represented by its code (e.g. code in Unicode encoding). You also need to store the information on where the string ends, it can be done by storing the length of the string.

f) What is the purpose of variables in programs?
Variables represent some storage location (memory cells) and can be used to store and retrieve data from this location. The type of the variable tells us what kind of data we can store in the corresponding memory.

g) What is the meaning of this operator in Java: ||
Logical OR

h) What does a compiler do? Why do we need to compile programs?
Translates high-level programming code into low-level instructions. Processor executes low-level machine instructions only.

i) What is the Java standard library?
A collection of Java classes designed for various purposes.

j) How would you store information about person’s eye color in a Java program? Are there any other alternatives?
We can encode each color as a separate number, 0 – green, 1 – gray, ... Alternatively, we can just store a color as a string. There can be other alternatives...
### Problem 2 – Java expressions (20pts)

For each row of the table, state whether the expression is correct or has an error. If there is no error, give the type and result of the expression. If there is an error, leave the type as blank and in the last column describe the reason of error.

Evaluate each statement with the original values of the variables. (Assume each row to be independent, that is result of some row execution doesn’t affect other rows).

**Hints:**
- consult with the Java operator precedence table
- note, that Math.log method has the following signature:

```java
public static double log(double a)
```

**Original values of the variables:**
- `int i1 = 2, i2 = 5, i3 = 7;
- `long L1 = 2, L2 = 5, L3 = 7;
- `boolean b = true;
- `double d1 = 2.0, d2 = 5.0, d3 = 7.0;
- `char c1 = '2', c2 = '5', c3 = '7';
- `String s1 = "2", s2 = "5", s3 = "7";`

<table>
<thead>
<tr>
<th>Statement</th>
<th>Error?</th>
<th>Type (if no error)</th>
<th>Reason of error or Result value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c3 – ’0’) * (c2 – ’0’)</td>
<td>No</td>
<td>int</td>
<td>35</td>
</tr>
<tr>
<td>i1 / i2 * i3 / i2</td>
<td>No</td>
<td>int</td>
<td>0</td>
</tr>
<tr>
<td>i3 = i2 * Math.log(i1)</td>
<td>Yes</td>
<td></td>
<td>Result of the right-hand side part is of type double, cannot assign double to integer - loss of precision</td>
</tr>
<tr>
<td>L2 / L3 == 0 &amp;&amp; c2 &gt; c1</td>
<td>No</td>
<td>boolean</td>
<td>true</td>
</tr>
<tr>
<td>L3 % L1 * L2 + 5 / 2 * i1</td>
<td>No</td>
<td>long</td>
<td>9</td>
</tr>
<tr>
<td>i1=(int)(1.0 * i2 / i1 + L2 / L1)</td>
<td>No</td>
<td>int</td>
<td>4 (note, 1.0 *i2/i1 evaluates to double value 2.5, but L2/L1 is an integer value 2)</td>
</tr>
<tr>
<td>1.0-0.1+0.1+0.1+0.1+0.1+0.1 == 0.5</td>
<td>No</td>
<td>Boolean</td>
<td>false</td>
</tr>
<tr>
<td>Integer.parseInt(s1 + s2) - i1</td>
<td>No</td>
<td>int</td>
<td>23</td>
</tr>
<tr>
<td>Integer.ToString(i3) + s3</td>
<td>No</td>
<td>String</td>
<td>&quot;77&quot;</td>
</tr>
<tr>
<td>++L1 + 2 == L2++ &amp;&amp; (b</td>
<td></td>
<td>!b)</td>
<td>No</td>
</tr>
</tbody>
</table>
Problem 3 – Trace program (20 pts)

(10 pts) Look at the following Java program:

```java
public class Problem3a {
    public static void main(String[] args) {
        java.util.Scanner in = new java.util.Scanner(System.in);
        String str = in.nextLine();
        String test = in.nextLine();
        String part1 = str.substring(0, test.length());
        String part2 = str.substring(str.length() - test.length(), str.length());

        boolean res1 = part1.equals(test);
        boolean res2 = part2.equals(test);

        if (res1 || res2)
            System.out.println("Yes");
        else
            System.out.println("No");
    }
}
```

Trace the execution of the program for the following inputs (2 pts each):

1) This is a test string
   result: Yes

2) abbbaaaacc
   bbb
   result: No

3) Hello World
   world
   result: No

What does the program accomplish? (4 pts)

It checks whether the first string starts or ends with the second string.
(10 pts) Look at the following Java program:

```java
public class Problem3b {
    public static void main(String[] args) {
        java.util.Scanner in = new java.util.Scanner(System.in);

        int number = in.nextInt();
        boolean ok = true;

        while (number != 0) {
            int tmp = number % 10;
            if (tmp % 2 == 1) {
                ok = false;
                break;
            }
            number /= 10;
        }

        if (ok)
            System.out.println("Yes :)"');
        else
            System.out.println("No :( ");
    }
}
```

Trace the execution of the program for the following inputs (2 pts each):

1) 248
result: Yes :)

2) 186
result: No :

3) 135
result: No :

What does this program accomplish? (4 pts)
It checks whether the given number consists of even digits only. (Note, the loop goes over all the digits in the number, once we found a character that is odd, we exit the loop)
Problem 4 – Interval intersection (25 pts)

In this problem you need to write a Java program that reads 4 floating-point numbers, representing 2 intervals on a line and checks whether the intervals intersect or not (it should print either “Intervals intersect” or “Intervals do not intersect” at the end). The program should first read the coordinates of the left and right ends of the first interval and then of the second one.

For example:

Please enter left point of the first interval: 1.2 3.0
Please enter left and right point on the first interval: 2.5 5.76
The intervals intersect!

Here is an example of intervals: first interval [1.2, 3.0] and second interval: [2.5, 5.76]:

These intervals do intersect.

**Definition:** Intervals intersect if they contain at least one point in common. So, if one interval completely contains another one the intervals are considered intersecting. Also, intervals that share only one of the endpoints also intersect, for example:

And the intervals on the following picture do not intersect:

Please complete the program on the next page:
public class Problem4 {
    public static void main(String[] args) {
        java.util.Scanner in = new java.util.Scanner(System.in);
        System.out.print("Please enter the left point of the first interval:");
        double left1 = in.nextDouble();
        System.out.print("Please enter the right point of the first interval:");
        double right1 = in.nextDouble();
        System.out.print("Please enter the left point of the second interval:");
        double left2 = in.nextDouble();
        System.out.print("Please enter the right point of the second interval:");
        double right2 = in.nextDouble();
        if (right1 >= left2 && right2 >= left1)
            System.out.println("Intervals intersect!");
        else
            System.out.println("Intervals do not intersect!");
    }
}
Problem 5 – Statistics (25 pts)

In this problem you need to write a Java program that asks user to enter a positive integer number N and then generates N random integers in the interval [-100, 100] (both numbers are included in the interval). To generate a random number in this interval you can use the following statement: (int)(Math.random() * 201) - 100;

Your program doesn’t need to print the generated numbers themselves. It should print some statistics calculated on these numbers only. The statistics are: average value, minimum and maximum.

Please complete the program on the next page:
public class Problem5 {
    public static void main(String[] args) {
        java.util.Scanner in = new java.util.Scanner(System.in);

        System.out.print("How many numbers do you want to generate? ");
        int N = in.nextInt();

        double average = 0;
        int min = 101;
        int max = -101;

        while (N > 0) {
            int number = Math.random() * 201 - 100;
            average += (double) number / N;
            if (min > number)
                min = number;
            if (max < number)
                max = number;
            N = N - 1;
        }

        System.out.print("Average = "+average+"; minimum = "+min+"; maximum = "+max);
    }
}