CSC 1350 Pre-Exam # 1

Section $\boxed{3/4}$

September 20, 2019

NAME: ______________________________

- Blue book is required. Fill in the information on the cover of your blue book and on the exam sheet.
- Answer Exercises 1, 4 and 6(A) on the exam sheet and all other exercises in your blue book.
- Calculators are not allowed.
- Use the back of the exam sheets if you need scratch paper.
- Read the instructions preceding each section carefully before beginning the section.
- Turn in the exam and your blue book before you leave.

DURATION: 50 Minutes

Table 1: Distribution of Points

PART	WORTH	SCORE
Exercises	100	/100

DO NOT TURN THIS PAGE UNTIL YOU ARE TOLD TO DO SO.

1 Exercises

Instruction: Read each question carefully before answering the question.

1. Mark the following identifiers either valid or invalid[10 points].

Identifier	Valid	Invalid
_1stYearGPA		
classOf2015		
player#		
dollar$		
GRAVITATIONAL-CONSTANT		

2. What are the values of the following expressions?[10 points] In each line, assume that

```
double x = 2.5;
double y = -1.5;
int m = 18;
int n = 4;
String s = "hello";
String t = "World";
```

(A) $x + n * y - (x + n) - y$

(B) $m/n + m\%n$

(C) $1 - (1 - (1 - (1 - (1 - n))))$

(D) s.replace(s.substring(0,1), s.substring(0,1).toUpperCase())

(E) t + " of " + t.substring(0,3) + t.substring(t.length() - 1,t.length())+"s"

3. Write the expressions below as functionally equivalent ones in the desired form in such a way that the number, order and type of operations are preserved.

(A) Write the Java expression

```
dm = m * (Math.sqrt(1+v/c) / Math.sqrt(1-v/c) - 1);
```

as a functionally equivalent expression in mathematical notation. [8 points]

(B) Write the mathematical expression $FV = PV\left(1 + \frac{rate}{100}\right)^t$ as a functionally equivalent Java expression. FV denotes the future value, PV, the present value, $rate$, the annual interest rate paid, and t, the number of years over which the investment is held. [7 points]

4. What is the output of the following code segment? Write the output in the chart provided, leaving an empty box to denote a whitespace. (Write only one character per box.)[10 points]

```
String name = "McKenzie";
int month = 9, day = 23, year = 2013;
double salary = 5035;
System.out.printf("Today's date is %02d/%02d/%d.%n",month,day,year);
System.out.printf("%5s earned $%.2f.%n",name,salary);
```

5. Assume *first* and *second* are integer variables.

```
if (first > second && first > third)
   System.out.printf("%d is the largest.",first);
else if (second > third)
   System.out.printf("%d is the largest.",second);
else
  System.out.printf("%d is the largest.",third);
```

(A) What will be the output of the code segment when $first = 8$, $second = 7$ and $third = 8$? [5 points]

(B) At least how many relational expressions will be evaluated when this code segment is executed? Give values for the variables $first$, $second$ and $third$ when this occurs. What would be the output of the code segment? [5 points]

(C) At most how many relational expressions will be evaluated when this code segment is executed? Give values for the variables $first$, $second$ and $third$ when this occurs. What would be the output of the code segment? [5 points]

(D) Explain why the code segment is inefficient by giving values for the variables $first$, $second$ and $third$ for which a relational expression is evaluated unnecessarily. [5 points]

(E) Calculate E, the average number of relational expressions evaluated per branch of this code segment assuming each branch is equally likely to be entered. [5 points]

$$E = \frac{Total\ Number\ of\ relational\ Expressions\ Evaluated}{Number\ of\ Branches} \tag{1}$$

(F) Write an optimized version of the code segment that does not unnecessarily evaluate a relational expression. [5 points]

(G) Calculate the average number of relational expressions evaluated per branch in the optimized version of the code segment that you wrote in 5.(F), again assuming the each branch is equally likely to be entered. [5 points]

6. Suppose A, B and C are boolean variables.

   (A) Complete the truth table below showing the logical value of each expression, where 0 denotes *false* and 1 denotes *true*. [5 points]

A	B	C	!B	!C	B||C	!(B||C)	A&&!(B||C)	A&&!B&&!C
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

Table 2: A Truth Table

   (B) What is the value of the expression $A \,\&\&\, !(B \,||\, C)$ when A is *true*, B is *false*, and C is *true*? [5 points]

   (C) Are $A \,\&\&\, !(B \,||\, C)$ and $A \,\&\&\, !B \,\&\&\, !C$ logically equivalent? [5 points]

2 Addendum

1. Assume the integer variables i, j, m, and n have been assigned the values $i = 6$, $j = 7$, $m = 11$, and $n = 11$.

```
System.out.print("Madam");
if (i < j)
{
   if (m != n)
      System.out.print("How");
   else
      System.out.print("Now");
}
System.out.print("I'm");
if (i >= m)
   System.out.print("Cow");
else
   System.out.print("Adam");
```

   (A) What will be the output of the code segment? [5 points]

   (B) Write a modified version of the code by adding *numRelExps*, an integer variable, and updating it where appropriate so that after the execution of this code segment its value is always the number of relational expressions evaluated during the execution of the code segment. Also, add code to print the statement "Number of Relational Expressions Evaluated: X", on the last line of output, where X is the number of relational expressions evaluated during the execution of the code segment. [8 points]

2. Assuming *firstNum* and *secondNum* are integer variables that have been declared and initialized, write a code segment that prints "*The product of the numbers is negative*", "*The product of the numbers is positive*" or "*The product of the numbers is zero*", when their product is negative, positive or zero, respectively. The code segment should evaluate between one and four relational expressions to give the correct output. No arithmetic operation or additional variable should be used in the code segment. Give values for *firstNum* and *secondNum* for which your code segment evaluates exactly one, two, three and four relational expressions. [20 points]

3. Assume x, y and z are integer variables. Verify that the code segment below always gives the correct output, then answer the question below.

```
if (x == y && x == z && y == z)
   System .out . println ("x, y and z are equal.");
else if (x == y && y != z && x != z)
   System .out . println ("Only x and y are equal.");
else if (x != y && y == z && x != z)
   System .out . println ("Only y and z are equal.");
else if (x != y && y != z && x == z)
   System .out . println ("Only x and z are equal.");
else
   System .out . println ("x, y and z are distinct. ");
```

   (A) What will be the output of the code segment when $x = 8$, $y = 8$ and $z = 7$? [5 points]

   (B) At least how many relational expressions will be evaluated when this code segment is executed? Give values for the variables x, y and z when this occurs. What would be the output of the code segment? [5 points]

   (C) At most how relational expressions will be evaluated when this code segment is executed? Give values for the variables x, y and z when this occurs. What would be the output of the code segment? [5 points]

   (D) Explain why the code segment is inefficient by giving values for the variables x, y and z for which a relational expression is evaluated unnecessarily. [5 points]

   (E) Calculate the average number of relational expressions that are evaluated per branch of the code segment above. [5 points]

   (F) Write an optimized version of the code segment that does not make an unnecessary evaluation of a relational expression during its execution. [5 points]

   (G) Calculate the average number of relational expressions that are evaluated per branch of the optimized version of the code segment that you wrote in 3.(F). [5 points]