CSC 1350 Exam # 1

Section 3/4

October 1, 2019

NAME: Kha Le

- Blue book is required. Fill in the information on the cover of your blue book and on the exam sheet.
- Complete Table 2 and Table 3 in the spaces provided on the exam sheet and answer 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: 80 Minutes

Table 1: Distribution of Points

EXERCISE	WORTH	SCORE
A	20	17
B	20	20
C	20	20
D	40	38
$\sum_{i=A}^{E} i$	100	/100

95%

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

Exercises

Instruction: Read each question carefully before providing an answer.

A. Consider the mathematical expressions below.

10

(a) Give an equivalent Java™ expression that preserves the order, number and type of arithmetic operations for each mathematical expression. Use the Java™ math library square root method for square roots and power method for exponentiations. [10 points]

i. $u = \frac{2a}{b} - \frac{b}{2a}$

ii. $z = \left(\frac{\sqrt{x+y}}{x^n + y^n - 1}\right)^n$

7

(b) Give the value of each expression where m and n are floating-point variables and x and y are integer variables with values 2.0, 3.0, 4 and 6, respectively. [10 [points]]

i. $(x + 5)\ \%\ 3\ +\ y\ *\ 2$

ii. $(y/x\ +\ x * m)\ *\ (y\ \%\ x)$

B. Consider the code segment in Listing 1:

Listing 1: Code Segment

```
1  Scanner cin = ________________;
2  System.out.print("Enter a number→");
3  double x = cin.__________;
4  System.out.print("x^3 = ");
5  double xCubed = x * x * x;
6  System.out.printf("____%n",xCubed);
```

6

(a) Provide the missing code for the three blanks that enables the user to enter a number from the keyboard so that the code segment outputs the cube of the number to the nearest ten thousandths (four decimal places). [6 points]

14

(b) Using the line numbers, give two other valid alternate execution sequences for the code segment that give the same interactivity and output. [14 points]

C. In the exercises below, assume all variables are Boolean.

(a) Consider the incomplete truth table below where 0 denotes false and 1 denotes true.

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

Table 2: A Truth Table

10

i. Complete the truth table shown in Table 2. [5 points]

ii. Are the expressions !(A && B) and !A && !B logically equivalent? Why or why not? [5 points]

10

(b) The *XNOR* logic gate *A XNOR B*, read "A exclusive nor B", is ***true* when both A and B are *true* or when both A and B are *false***. Otherwise, it is *false*. Using the variables *A* and *B*, the Boolean operators &&, || and !, and possibly parentheses, write a logically equivalent *Java* Boolean expression for the XNOR logic gate. [10 points]

D. Consider the code segment below assuming that the variables x and y are integers.

Listing 2: Code Segment

```
if (x == 0 ||  y == 0)
   System.out.println("(x,y) is on an axis.");
else if (x > 0 && y > 0)
   System.out.println("(x,y) is in the 1st quadrant.");
else if (x < 0 && y > 0)
   System.out.println("(x,y) is in the 2nd quadrant.");
else if (x < 0 && y < 0)
   System.out.println("(x,y) is in the 3rd quadrant.");
else
   System.out.println("(x,y) is in the 4th quadrant.");
```

(a) What is the maximum number of relational expressions that would be evaluated when the code segment in Listing 2 is executed? Give values for the variables, using integers between -5 and 5, when this occurs. [8 points]

(b) Explain why the code segment in Listing 2 is a suboptimal way to determine whether the point (x, y) lies on an axis or the quadrant in which it lies in the Cartesian coordinate plane. Then give values of the variables, using integers between -5 and 5, that support your explanation. Be very precise about the inefficiency by giving the relational expression that is evaluated unnecessarily. [8 points]

(c) Suppose $\overline{E}$ denotes the average number of relational expressions that are evaluated per branch of a code segment involving *if*-statement(s). Complete the chart below and then calculate $\overline{E}$ for the code segment in Listing 2. [8 points]

# of Relational Expressions	Branch of If-Statement
11 2	1st (line 2)
4	2nd (line 4)
5	3rd (line 6)
7	4th (line 8)
6	5th (line 10)

-1

$\frac{2+4+5+7+6}{5} = \frac{6+12+6}{5} = \frac{24}{5} = 4.8$

Table 3: Relational Expressions Evaluated Per Branch

(d) Give an optimal version of the code segment in Listing 2 that eliminates the potential for unnecessary evaluations of relational expressions in the code segment in Listing 2 while keeping the *println* statements in the same relative order and without the use of any new variable or arithmetic operation. [8 points]

(e) Drawing and completing a chart similar to the one given in Table 3, calculate $\overline{E}$ for the optimal version of the code segment that you wrote in D.(d). [8 points]

$\frac{2+4(4)}{5} = \frac{2+16}{5} = \frac{18}{5} = 3.6$