CSC 1350 Exam # 2

Section (3/4)

October 29, 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	20
B	20	20
C	20	18
D	40	40
$\sum_{i=A}^{E} i$	100	98/100 + 1

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

99 W.E.D

revised

11/6

Exercises

Instruction: Read each question carefully before providing an answer.

A. Fill in the blanks in exericse (a) to complete the declaration for a one-dimensional array of integers called *list* containing 2, 0, 1, 3 and 0 in the order listed. Then give the value of each expression in exercises (b) through (e) or indicate that the expression is indeterminate.

(a) _____ *list* = _____; [4 points]

(b) list[2] + list[3] [4 points]

(c) list[list[0] + list[1]] [4 points]

(d) list[list[3] + 1] [4 points]

(e) list[list[list[4]+list[2]]] [4 points]

B. Give the output that the program in Listing 1 produces when it is compiled and executed. [20 points]

Listing 1: A Java Program

```
public class Enigma
{
   public static void main(String[] args)
   {
      int i, j;
      for (i = 1; i <= 4; i++)
      {
         for (j = 1; j <= 7; j++)
         {
            if (j >= i && j <= 8-i)
               System.out.print(9 - 2*i);
            else
               System.out.print(" ");
         }
         System.out.println();
      }
   }
}
```

C. $H = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots$ is called the harmonic series. Consider the starter code for the static method in Listing 2.

Listing 2: A Static Method

```
/**
 * Computes, up to the specified number of terms, the
 * Harmonic series
 * @param n the number of terms
 * @return the sum of the harmonic series consisting of the
 * first n terms  or 0 when n is less than 1.
 */
public static double harmonicSeries(int n)
{
   //provide the implementation for this method
}
```

(a) Provide the missing code for the Java™ method in Listing 2 so that it computes the harmonic series up to its first n terms. [16 points]

(b) Write a Java™ statement that calls the method and prints the sum of the harmonic series up to its first 50 terms. Assume that this statement is in the same class as the method. [4 points]

D. Consider the code segments below and provide answers for the exercises.

Listing 3: Code Segment

```
1 for (int i = 1; i < 6; i++)
2 {
3    System.out.printf("%d ",5*i - 2);
4 }
```

(a) How many iterations will the code segment in Listing 3 make and what will it output? [10 points]

(b) Tally the number of times the operations in Table 2 are carried out when the code segment in Listing 3 is executed by completing Table 2. [10 points]

Operators	Total Number of Times Executed
Add/Subtract	
Multiply/Divide/Modulus	
Comparison	

Table 2: Tally of Basic Operations

(c) Complete the code segment in Listing 4 so that it produces the same output as the code segment in Listing 3. For the blank on line 1, give the most-restrictive bound for the loop counter. How many iterations will the completed code segment make? [10 points]

Listing 4: Code Segment

```
1  for (int i = 7; i ___; i+=5)
2  {
3     System.out.printf("%d ",____);
4  }
```

(d) Complete the chart shown in Table 3. The chart should show the tally of the operations when the completed version of the code segment in Listing 4 is executed. Which code segment is more efficient and why? [10 points]

Operators	Total Number of Times Executed
Add/Subtract	
Multiply/Divide/Modulus	
Comparison	

Table 3: Tally of Basic Operations