CS 3 Test 1 study guide Summer

All chapters 1- 8 are included on the test.

- Expressions/data types
- Decision making (if, switch, conditional statement)
- Loops (while, for, do-while, for-each)
- Arrays
- Methods / functions
- Classes (system : String, Scanner, …)
- Driver classes (includes main method)

To practice do the problems at the back of each section and chapter, review your labs and assignments. Here are some sample questions as well:

1. **Expressions**
   For each expression in the left-hand column, indicate its value in the right-hand column. Be sure to list a constant of appropriate type (e.g., 7.0rather than 7for a double, Strings in quotes, true/falsefor a boolean).

Expression	Value
3 * 4 + 5 * 6 + 7 * -2	_____
1.5 * 2.0 + (5.5 / 2) + 5 / 4	_____
"1" + 2 + 3 + "4" + 5 * 6 + "7" + (8 + 9)	_____
1 / 2 > 0 || 4 == 9 % 5 || 1 + 1 < 1 - 1	_____

2. **Parameter Mystery**

```
public class ParameterMystery{
    public static void main(String[] args){
        String y ="tyler", z = "tv", rugby = "hamburger" ,java = "donnie",  x = "java";

        hamburger(x, y, z);
        hamburger(z, x, y);
        hamburger("rugby", z,java);
        hamburger(y, rugby, "x");
        hamburger(y, y, "java");
    }

    public static void hamburger(String y, String z, String x)
    {
        System.out.println(z + " and " + x + " like " +y);
    }
}
```

3. **If/Else**
For each call of the method below, write the value that is returned:

```
public static int mystery(int a, int b) {
        int          c;
        if (a > b)
              c = a;
        else if (b % a == 0)
              c = b;
        else
             c = b + (a - (b % a));
        return c;
}
```

Method Call	***Value Returned***
mystery(4, 2)	_____
mystery(5, 4)	_____
mystery(5, 13)	_____
mystery(5, 17)	_____
mystery(4, 8)	_____

4. **While Loop** For each call of the method below, write the output that is printed:

```
public static void mystery(int i, int j) {
   while (i != 0 && j != 0) {
            i = i / j;
            j = (j - 1) / 2;
            System.out.print(i + " " + j + " ");
      }
       System.out.println(i);
}
```

Method Call	***Output***
mystery(5, 0);	_____
mystery(3, 2);	_____
mystery(16, 5);	_____
mystery(80, 9);	_____
mystery(1600, 40);	_____

5. Trace the following code segment and show what is printed on the monitor:

```
public static int mystery(int x) {
        int y = 1 , z = 0;
        while (y <= x) {
              y = y * 10;
              z++;
        }
        z--;
        System.out.println(y + z + x);
        return z;
}
int     a = 40 , b;

b = mystery(a);
System.out.print(a + b + (a+b) );
```

6. What is the output of the following code segment?

```
String[]   students = new String[10];
String studentName = "Peter Smith";
students[0] = studentName;
studentName = "joe";
System.out.println(students[0] + "\t" + studentName);
```

Programming sample questions:

7. Write a static method named hasMidpointthat accepts three integers as parameters and returns trueif one of the integers is the midpoint between the other two integers; that is, if one integer is exactly halfway between them. Your method should return falseif no such midpoint relationship exists. The integers could be passed in any order; the midpoint could be the 1st, 2nd, or 3rd. You must check all cases.

Calls such as the following should return true:	Calls such as the following should return false:
hasMidpoint(4, 6, 8)	hasMidpoint(3, 1, 3)
hasMidpoint(2, 10, 6)	hasMidpoint(1, 3, 1)
hasMidpoint(8, 8, 8)	hasMidpoint(21, 9, 58)
hasMidpoint(25, 10, -5)	hasMidpoint(2, 8, 16)

8. Write a static method named sequenceSum that prints terms of the following mathematical sequence:

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots + \frac{1}{N} \quad \textit{also written as:} \quad \sum_{i}^{N} \frac{1}{i}$$

Your method should accept a real number as a parameter representing a limit, and should add and print terms of the sequence until the sum of terms meets or exceeds that limit. For example, if your method is passed 2.0, print terms until the sum of those terms is at ≥ 2.0. You should round your answer to 3 digits past the decimal point.

The following is the output from the call sequenceSum(4); 1+ 1/2 + 1/3 + 1/4 = 2.083

Calls	sequenceSum(0);	sequenceSum(1);	sequenceSum(2);
Output	0.0	1 = 1.000	1 + 1/2 = 1.500
Call	sequenceSum(8);		
Output	1 + 1/2 + 1/3 + 1/4 +1/5 + 1/6 + 1/7 + 1/8 = 2.718		

9. Write a java code segment to declare an array of size 10 of type String and read data for them from keyboard.

10. Write a java code segment to declare an array of size 10 of type String and read data for them from a file, prompt user for the file name. Data is stored in a file with an address per line.