CS 170 ch.5 Lab 2

# Task 1

Question: **Value-returning methods**

Students will write a class called Dimensions that will:

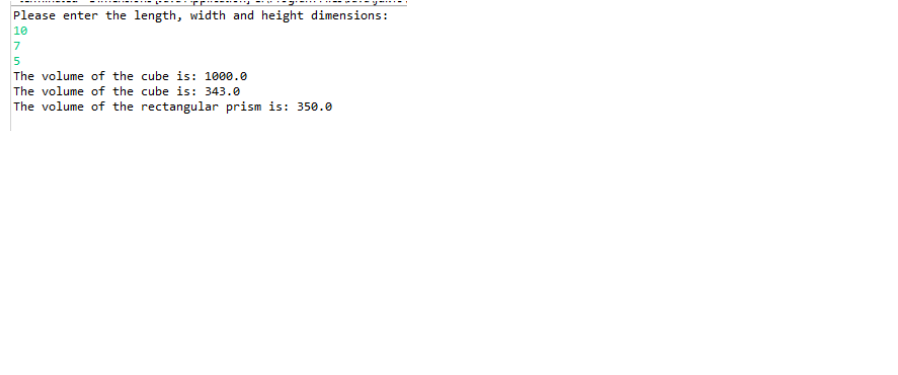
* ask the user to input the length, width, and height dimensions
* write a method cubeVolume that calculates the volume of a cube by passing in only one argument of the dimensions from above. The Formula needed is V = a^3
* write a method that calculates the volume of a rectangular prism called rectangularPrismVolume by passing in three arguments. The formula needed is V= w \* h \* L
* output the volume for the following test cases with an appropriate message.

Test Cases:

   #1        calculate the cubeVolume  of length  10

   #2        calculate the cubeVolume  of width  7

   #3        calculate the rectangularPrismVolume  of length  10, width  7    height   5



|  |
| --- |
|  |

|  |
| --- |
|  |

# Task 2

Question: **Simple Calculator: Value Returning methods**

**Part A.**

Write a class SimpleCalculator. java that mimics the functions of a simple calculator.

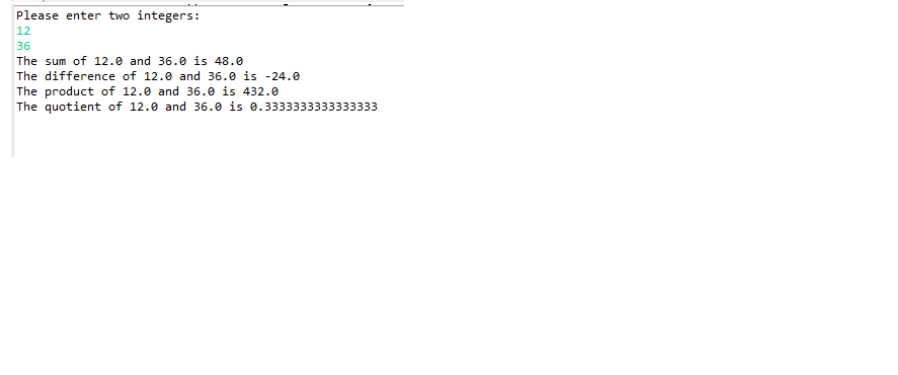
Your code should contain the following methods:-

      main, addition, subtraction, multiplication, and division.

In the main method, let the user enter any two integers.  Invoke all 4 methods by passing in the two numbers as your arguments.

Design your  output messages so that they reflect all the simple calculator operations and their results ( print the two numbers and each operation result)

**(Note: try invoking your methods from inside your print statements, this is much more efficient.)**



|  |
| --- |
|  |

|  |
| --- |
|  |

**Part B.**

Adding decisions to a method (if statements)

In this part, do not invoke the subtraction method from the main method, but rather from inside the addition method.

In the subtraction method check if the first number is larger than second, if it is then subtract but if not then swap the subtraction. The result should be positive.

Do not invoke the division method from the main method, but rather from inside the multiplication method.

In the division method check to see if the first number is larger than the second, if it is then divide, but if not then swap the division. The result should not be a decimal value (less than 1). Your code should also avoid a division by zero exception.

Note Use the same test case in part A, but the results should be different for subtraction and division.

|  |
| --- |
|  |

|  |
| --- |
|  |

# Task 3

Question: **Recursion**

Students will implement recursion (a method calls itself) by writing a class called

RecursiveTriangles.java that contains:

1.a main method: ask the user to input the length (sides of a triangle)  of a triangle.

                                     invoke the PrintTriangles() method by passing in the length as an

                                      an argument.

2.a method PrintTriangles( int sideLength)

 This method is a recursive method that calls itself multiple times.

  (refer to ch.5 slides for this method's code)



|  |
| --- |
|  |

|  |
| --- |
|  |