

Problem:

In your own words, analyze and explain the program you implemented below, using just one or two sentences, what it does. Write this explanation as a comment at the top of your program. **This explanation is part of your grade (Look at the rubric).** Do not be too detailed and/or vague (For example, do not write: it opens a file, then declares variables, and then reads from a file...); it should look like the problem descriptions in my previous lab assignments.

Requirements:

Implement in C++ the algorithm solution (Pseudo-code) shown below:

- 01) Include your libraries here. Explain the libraries being used with comments.
- 02) Create value-returning function prototype **half()**.
Parameter(s): Number to half
- 03) Create value-returning function prototype **square()**.
Parameter(s): Number to square
- 04) Declare variable(s) named "numA", "numB", and "numC" that hold(s) **double-precision real number(s)**.
- 05) Declare variable(s) named "halfValueA", "halfValueB", and "halfValueC" that hold(s) **double-precision real number(s)**.
- 06) Declare variable(s) named "squaredValueA", "squaredValueB", and "squaredValueC" that hold(s) **double-precision real number(s)**.
- 07) Prompt the user:
Please enter three (3) values (Add a space between each):
- 08) Get the three (3) values from the keyboard and store it in the corresponding variable.
- 09) Call the **value-returning function half() three (3) times** and assign to the corresponding variables.
Parameter(s): Number to half
- 10) Call the **value-returning function square() three (3) times** and assign to the corresponding variables.
Parameter(s): Number to square
- 11) Format output in fixed decimal notation displaying two digit(s) after the decimal point.
- 12) Display on the screen the message: ⁱ

[halfValueA], [halfValueB], [halfValueC] is half of [numA], [numB], and [numC]

[squaredValueA], [squaredValueB], [squaredValueC] is half of [numA], [numB], and [numC]

Press any key to continue...
- 13) Terminate your program with integer 0
- 14) Create **value-returning function definition for half() that passes one (1) parameter** and returns the half of the passed value.
Requirements and algorithm for the function
01) Calculate and return the half of the value(s) ⁱⁱ
Formula: $x_A / 2.0$
Parameter(s): Number to half
- 15) Create **value-returning function definition for square() that passes one (1) parameter** and returns the square of the passed value.
Requirements and algorithm for the function
01) Calculate and return the square of the value(s) ⁱⁱⁱ
Formula: $x_A * x_A$
Parameter(s): Number to square

Your program must compile without errors or warnings. Try to work on it before we meet for the lab so you can finish it during the lab period.

I am posting the executable of my solutions for your reference. Please run them and ensure that your programs works like mine. Try different numbers and compare the results returned by your solution and mine.

Your program must have the following comments at the top. Don't forget to include them because they will count toward the grade of this lab.

```
//*****  
// Course Number and Section:  
// Course Semester:  
// Your Name:  
// --Only if he helped you with the assignment--  
// Teammate Name:  
// -----  
// Program Description:  
//  
//*****
```

Submission:

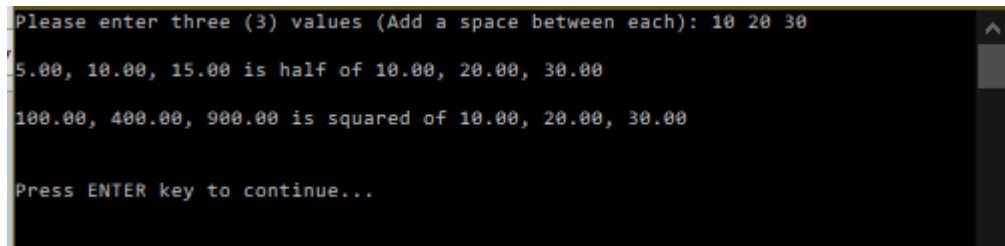
Please name your file **lab3TXX** (where **XX** is the team/group number). If you work with a teammate, then make sure to include them on the lab heading.

DO NOT include blank spaces in the name of the file please.

For example: lab01T00.cpp

When done, submit your solution through Blackboard using the “Assignments” tool. Do Not email it.

Screenshots:



```
Please enter three (3) values (Add a space between each): 10 20 30  
5.00, 10.00, 15.00 is half of 10.00, 20.00, 30.00  
100.00, 400.00, 900.00 is squared of 10.00, 20.00, 30.00  
Press ENTER key to continue...
```

Grade Criteria: ^{iv}

	Levels of Achievement			
Criteria	Exceptional	Proficient	Satisfactory	Unsatisfactory
Heading	100.00%	50.00%	25.00%	0.00%
Weight 10.00%	Included program heading and information is fully completed.	Included program heading, but is missing information.	Included program heading, but did not fill information.	Did not include program heading.
Documentation	100.00%	50.00%	25.00%	0.00%
Weight 15.00%	Program is fully commented, comments are placed above the statements and comments are your own words.	More than half comments added to program, comments are placed above the statements, and/or comments are your own words.	Less than half comments added to program, comments not placed above statements and/or comments are not your own words.	No comments added to program.
Organization	100.00%	50.00%	25.00%	0.00%
Weight 20.00%	White space is applied appropriately and program follows a structured organization. Ex: Follows algorithm and/or some pattern.	White space is mostly added and follows mostly structured organization. Ex: Follows algorithm and/or some pattern.	White space is minimally added and follows minimally structured organization. Ex: Follows algorithm and/or pattern.	No white space is added and follows no structured organization. Ex: Follows algorithm and/or pattern.
Requirements	100.00%	50.00%	25.00%	0.00%
Weight 35.00%	Program requirements were fully met. Look at program handout requirements.	More than half of the program requirements were met. Look at program handout requirements.	Less than half of the program requirements were met. Look at program handout requirements.	Program requirements were not met. Look at program handout requirements.
Accuracy	100.00%	50.00%	25.00%	0.00%
Weight 20.00%	Program output is accurate and is formatted appropriately. Ex: Spacing, new lines, etc.	Program output is accurate and/or mostly formatted appropriately. Ex: Spacing, new lines, etc.	Program output is not accurate and/or mostly formatted appropriately. Ex: Spacing, new lines, etc.	Program output is not accurate and is not formatted appropriately. Ex: Spacing, new lines, etc.

- i Pay attention to the new lines when displaying the message. Look at screenshot and/or sample executable program.
- ii Be careful for mixed datatype(s)
- iii Be careful for mixed datatype(s)
- iv More points may be lost for other reasons not listed in the criteria rubric and/or requirements.