**CS 10 - Assignment 7: Mathematics functions**

**Collaboration Policy**

We encourage collaboration on various activities such as lab, codelab, and textbook exercises. However, **no collaboration between students is allowed on the programming assignments**. Please be sure to read and understand our full policy at: [Full Collaboration Policy](https://docs.google.com/document/d/1WyzL3qvKLrC1UCRf178b_wYWQmEZlhDObFNFb79U63I/edit?usp=sharing)

**Submission Instructions**

Submit to [R’Sub](https://galah.cs.ucr.edu) testing, feedback and grading.

**Assignment Specifications**

This exercise is all about designing functions to be as broadly useful as possible: providing them with the information they need to do their job (via parameters, either by value or reference); and having them hand back their results (either via the reference parameters or the return statement).

**Your Assignment**

Write a program that allows the user to select a mathematical operation to solve. Each operation will be implemented as a function; a single "helper" function will gather the operands needed for the selected operation. Some of these operations can "break" under certain conditions (e.g. divide by 0), so the functions must be able to flag any errors by returning an error code (we provide global const int variables for these codes).

**Develop this program one function at a time, starting with the "helper" function for operand input; for each function, write a "test harness" that tests it for a variety of inputs *(but do not submit these individual test harnesses)*.**

**The functions**

1. acquireOperands: 1 const reference parameter (requested operation), 3 reference parameters (for up to 3 operands);   
   returns number of operands acquired.
2. divsion: calculates a/b  
   2 value parameters (for operands), 1 reference parameter (for result);  
    returns error code meaning "OK" or "Divide by 0"
3. Pythagorean equation: hypotenuse c = sqrt(a2 + b2)  
   2 value parameters (for operands), 1 reference parameter (for result);   
   returns error code meaning "OK" (no possible errors)
4. quadratic equation: solves ax2 + bx + c = 0  
   x=\frac{-b\pm\sqrt{b^2-4ac\ }}{2a}.  
    3 value parameters (for operands), 2 reference parameters (for two roots);  
   returns error code meaning "OK", or "Divide by 0", or "Negative Discriminant"

**Notes:**

* acquireOperands must request *ONLY* the number of inputs required for the operation
* the operation functions do *NOT* "return" their results to main: results are communicated via the reference parameters. The return value is used for the error codes.
* there is *NO* input or output in any of the operation functions - all i/o is done in main and the acquireOperands function.
* you will need to #include <cmath>

**main()**

Once you have your functions written and tested, you can write main:

* Prompt the user to select an operation.   
  If the choice is anything other than "division" or "pythagorean" or "quadratic", output a message "Operation not supported", and prompt again.
* Once you have a valid operation, invoke acquireOperands to acquire the correct number of operands for the operation *(you will capture this number into a variable in main, but you won't actully use that variable for anything in this assignment)*.
* now you can invoke the appropriate operation function, and display the requested operation (Equation: ), and either the result(s) (Result: ) or the appropriate error message (Error: ), as per the examples.

**Function prototypes & headers:**

/// @brief acquire the proper number of numeric inputs based on operation string

///

/// The numeric inputs are set into x, y, and z in that order;

/// the number of numeric inputs acquired is returned to the caller.

/// Not all operations require all three values:

/// do not set operands that are not needed.

///

/// @param op the operation to be performed such as division

/// @param x the first numeric input

/// @param y the second numeric input

/// @param z the third numeric input

/// @return the number of numeric inputs that were acquired

int acquireOperands(const string & op, double & x, double & y, double & z);

/// @brief calculate quotient for provided values (a/b); returns error code

///

/// @param a the dividend of the equation

/// @param b the divisor of the equation

/// @param result reference to place the quotient in

/// @return returns the integer representing the state of the calculation

/// using global constants for OK and DIV\_ZERO

int division(double a, double b, double &result);

/// @brief calculate the roots to the quadratic formula for a polynomial of

/// the form a\*x^2 + b\*x + c = 0; returns error code when necessary.

///

/// @param a the coefficient of x^2 in the polynomial equation

/// @param b the coefficient of x in the polynomial equation

/// @param c the last value of the polynomial equation

/// @param root1 reference for the first root of the quadratic formula

/// @param root2 reference for the second root of the quadratic formula

/// @return returns the integer representing the state of the calculation

/// using global constants for OK, DIV\_ZERO, and SQRT\_ERR

int quadratic(double a, double b, double c, double &root1, double &root2);

/// @brief calculate c for the pythagorean theorem a^2 + b^2 = c^2

///

/// @param a the value of a in the equation

/// @param b the value of b in the equation

/// @param c reference for the hypotenuse value

/// @return returns the integer representing the state of the calculation

/// using global constant for OK

int pythagorean(double a, double b, double &c);



**Example Runs** (User input has been **bolded and underlined** for emphasis.)

user@cs10\_assignments$ g++ mathematics.cpp -o math.out

|  |  |
| --- | --- |
| user@cs10\_assignments$ run math.out  Please choose an operation: **quadratic**  Enter your first number: **1**  Enter your second number: **0**  Enter your third number: **-4.84**  Equation: 1x^2 + 0x + -4.84 = 0  Result: -2.2, 2.2  user@cs10\_assignments$ | user@cs10\_assignments$ ./math.out  Please choose an operation: **division**  Enter your first number: **1.5**  Enter your second number: **0**  Equation: 1.5 / 0  Error: Cannot divide by zero.  user@cs10\_assignments$ |