SY DE 121
Lab Number 6

Notes:

1) Remember to review the SD121 Style Guide to know how to properly document your function declarations. Note that the Style Guide refers to "declarations" as "prototypes". Both terms are acceptable, but Savitch uses "declarations".
2) Now that we have started function-based coding, you are allowed to use carefully selected global variables e.g., you will probably need an expression for pi in this lab – this can be created as a global constant. In Lab#7, we will start putting such variables into our own header (*.h) file.

Exercise 1: Quadratic equation solver

Learning Objectives: Practice properly implementing a function.

Read This First

As you should already know, the solutions to the quadratic equation

$$ax^2 + bx + c = 0$$

are given by

$$\frac{-b+\sqrt{b^2-4ac}}{2a} \text{ and } \frac{-b-\sqrt{b^2-4ac}}{2a} \text{ and } \frac{-b-\sqrt{b^2-4ac}}{2a}$$

What to do

Write a program to compute the solutions of a quadratic equation. The user should be prompted (in a separate function) for the values of the coefficients *a, b,* and *c* and the data should be entered within that same function. Assume that the values *a*, *b*, and *c* are all real. The program should then compute and display the roots. You should be able to calculate and display the roots whether or not they are real or complex. Use separate functions for calculating the roots and for displaying the roots.

Note

This is a "messy" way of implementing a function to determine quadratic roots. In the course, this problem will be more easily implemented after you have learned about structs (later!). Structs will allow you to implement this solution by representing the complex numbers in a more concise manner. Implementing a clas (later!) is an even more appropriate means of representing complex numbers.

What To Do

Create a program that will:

(a) create a function that prompts the user to enter the values for the thre coefficients (no error checking required);

(b) create a function that performs the calculation for the solution to the quadrati equation; and

(c) create a function that display the roots

Exercise 2: Triangle side calculator

Learning Objective: More practice with functions!

Read This First

The cosine rule in geometry relates one angle in a triangle to the lengths of all thre sides. For any triangle where side c is opposite angle θ (theta) the equation is:

$$c^2 = a^2 + b^2 - 2ab\cos\theta.$$

In this exercise, you will write code to compute the angle, θ, given the length c each of the sides.

First check that the triangle sides are valid prior to performing any operation. A se of side lengths only makes sense if *all three sides are positive*, and if *no side i greater than or equal to the sum of the other two sides*.

What To Do

Create a program that will:

(a) create a function that prompts the user to enter the values for each of the side (no error checking);

(b) create a function that checks to verify that the three lengths can create a triangle and

(c) create a function that returns the angle opposite to length c (in degrees). Yo can use a trigonometric equality (i.e., use the inverse 'tan' function) as a mean of setting up a constant double for PI.

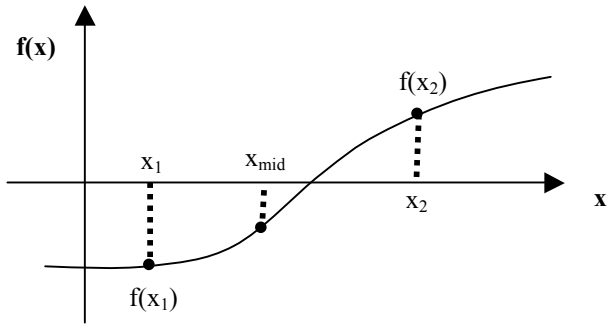
Exercise 3: Solving nonlinear equations: Bisection method

Learning Objectives: Build a function to solve nonlinear equations. Apply the method to a physics problem.

Read This First

Not all engineering problems are linear in nature. Nonlinear equations generally require dedicated algorithms to help solve.

The bisection method is one method that can be used to find roots for nonlinear equations. The idea is simple. Over some interval, the function is known to pass through zero because it changes sign. Evaluate the function at the interval's midpoint and examine its sign. Use the midpoint to replace whichever limit has the same sign. Graphically,



At this step, $\mathbf{x_{mid}}$ would become the new $\mathbf{x_1}$. Iterate in this manner until an appropriate solution is reached. Since the root is expected to be a real number without an exact digital representation, you must set some reasonable tolerance level that approximates the actual root. When the difference between one solution to the next is below this threshold, the appropriate approximation is assumed to be determined.

What to do

Write a program to solve the following non-linear equation for $\boldsymbol{\theta}$: $\tan(\boldsymbol{\theta}) - \sin(\boldsymbol{\theta}) = 1/12$.

For your program, you are welcome to hardcode the initial left and right boundaries i.e., no need for user input. Assume that the initial bounds are zero and 89 degrees (avoid 90 degrees because of potential problems with the tan function). Create tw functions: **one for calculating the nonlinear function** and **the other fo performing the bisection method**. In the bisection function, you should verify tha your bounds definitely surround a root (i.e., zero crossing). Back in the mai function, output the root to the screen. Also, perform a check to see if your solutio is a true root of the nonlinear function.

Exercise 4: Iterative and Recursive Functions

Learning Objectives: Learn how to implement a recursive function.

Read This First

The Fibonacci series begins with 0 and 1 and has the property that each subsequer Fibonacci number is the sum of the two previous Fibonacci numbers ie.

$$f(0) = 0,\ f(1) = 1,\ f(n) = f(n-1) + f(n-2)$$

What to do

Write an iterative C++ function that will determine the nth Fibonacci number. Alsc write a recursive C++ function to perform the same task. Call each of thes functions from the same main function to demonstrate that each produces the sam answer. Allow the user to enter *n* and make sure that $n >= 0$.

Answer the following questions:

For which method should it be easier to write the code? Why?
Which method is more computationally efficient? Why?

What To Hand In

You are only required to **submit three of the four** indicated problems for grading although you need to understand all of the problems in preparation for the midterm.

Ex1: lab0601.cpp

Ex2: lab0602.cpp

Ex3: lab0603.cpp

Ex4: lab0604.cpp and lab0604.txt

Due Date

Note different submission time for this lab!!! To give you some more flexibility in preparing for the upcoming midterm exam, this lab is due Monday October 23 by 9:00am.