

Learning Goals:

- learn to design and implement functions
- learn to use functions, including:
 - passing parameters to functions
 - getting return values from functions
 - writing functions that can be reused
- learn to incrementally build and test your code

What to hand in:

- Hand all of the following in using Git. For details on how to use Git, please see “How to submit labs and assignments” on the course page (csci.viu.ca/~carruths/Courses/CSCI160)
 - Lab3Prelab.pdf
 - Due night before your lab (by 23:59)
 - Lab3Checklist.pdf
 - Due: 23:59, three days following your assigned lab section (i.e. Tuesday Lab is due Friday, Thursday Lab is due Sunday)
 - Lab3.cpp
 - Due: 23:59, three days following your assigned lab section (i.e. Tuesday Lab is due Friday, Thursday Lab is due Sunday)
- Note:
 - late submissions will be penalized 20% per day, and will not be marked after 3 days in order that solutions can be posted in a timely manner
 - All work must be individual. Plagiarized work will result in a mark of 0. Further penalties may apply.

Marking Scheme:

- Prelab is completed on time – 1 mark
- All files meet given specifications and requirements – 6 marks
- Code readability and comments – 2 marks
- Self-assessment is consistent with quality of work – 1 marks

Part 1:

In lab, we will re-write the C++ program we wrote last week, moving the code to compute the volumes and areas of the shapes into functions.

We will also learn to incrementally build our code to help find and identify errors more quickly.

To start we will *prototype* the functions we need. We will write the following functions:

1. a function to output the description of the program
2. a function to get the height and radius of the tree
3. a function to compute and return the length of the tree
4. a function to compute and return the surface area of the tree
5. a function to print out the height, length, radius and surface area of the tree
6. a function to compute and output the length of fabric needed

At this point, if you try to compile your code, you will get an error because you've prototyped the functions but not defined them. You broke your promise!!

Defining functions:

Before we can use our functions, we need to define them. For each function we declared we will do the following:

- stub out the function
- add the function call to the main routine
- test our code
- complete the body of the function
- test our code

This iterative process lets us break up the errors we (might) find into different kinds of errors:

- errors declaring the function or in the function signature
- errors in the body of the function

Stubbing out functions:

Another trick we'll use is called *stubbing out functions*. We will now stub out both the functions for computing the Cube's values, and compile and run the program.

Once your code runs, we will finish implementing the stubbed-out functions. You can copy/paste the code from Lab 2, although be aware that you may need to fix the names of some variables so that they match what is expected inside the function.

Fill out body of functions:

Once you've tested that the function can be called, add the code to the body of the function that implements what the function is supposed to do. Then test again!

Calling functions

When we're done, the main function will do the following:

- declare the variables needed to hold the user input
- prompt the user for input
- call the functions to compute the surface area of the cube and assign the return values to a variable
- output the area
- call the functions to compute and output the volume of the cube

Part 2:

Your final job, before submitting your code, is to complete the portion for the length of fabric. Compile, run, and test your code.

Specifications:

Your lab3.cpp program will

- Describe what the program will do
- Meets of all specs for Lab2
- Use a single function to get the height and radius of the tree
- Use a function to compute and return the length of the tree

- Use a function to compute and return the surface area of the tree
- Use a function to print out the height, length, radius, and surface area of the tree
- Use functions to compute and output the length of fabric needed
- Terminate safely

Requirements:

- Code compiles and runs with no warnings
- Code terminates safely (no crashes)
- Code compiles with no errors

Code Readability and Comments:

- header comment includes:
 - name of author
 - name of program
 - brief description of purpose of program
- Brief comment for all variable declarations
- Brief comment for all logical blocks
- Comment for each function that *describes how it is used: input, output*
- File layout is:
 - header comment
 - includes
 - constant declarations
 - function prototypes
 - main routine
 - function implementations
- Layout:
 - correct placement of braces
 - no lines wrapping (approx. limit 100 characters)
 - correct indentation
- Correct naming conventions followed
- Appropriate use of constants
- Appropriate use of functions:
 - reusable functions
 - modular code
 - passing parameters
 - returning values

Bonus:

- Compute and output the difference in volumes between the two shapes.

Assumptions:

- No error checking is required
- All values entered by the user are positive (>0)