# Description

This lab accompanies Chapter 3 of *Starting Out with Programming Logic & Design*.

# Part 1: Questions

Complete the assignment in Topic 3: Lab A named, "Lab 3a Questions", in Canvas.

# Part 2: Activity

## Definitions

### Algorithm

An algorithm is a set of well-designed logical steps that must take place in order to solve a problem.

### Module

A Module is a group of statements that exists within a program for the purpose of performing a specific task.

Modules are commonly called procedures, subroutines, subprograms, methods, and functions.

The code for a module is known as a module definition. To execute the module, you write a statement that calls it.

The format for a module definition is as follows:

Module moduleName( )

Statement

Statement

Etc.

End Module

Calling a module is normally done from the main ( ) module such as:

Call moduleName( )

Generally, local variables should be used and arguments should be passed by reference when the value of the variable is changed in the module and needs to be retained. For example:

Module main( )

Declare Real number

Call inputData(number)

Call printData(number)

End Module

//accepts number as a reference so the changed value

//will be retained

Module inputData(Real Ref number)

Display “Input a real number: “

Input number

End Module

//number does not to be sent as a reference because

//number is not going to be modified

Module printData(Real number)

Display “The number is “, number

End Module

### Function

The code for a function is known as a function definition. To execute the function, write a statement that calls it. To create a function, write its definition. The keyword *void* is used before a function name, followed by parentheses. Here is the general format of a function definition in C++:

void functionName() // a function definition goes after main

{

statement

etc.

}

Calling a function is done in order to make the module execute:

functionName(); // calling this function

A function prototype is usually used in C++:

void functionName(); // a function prototype goes before main

Variables can either be local or global in scope. A local variable is created inside a function and cannot be accessed by statements that are outside a function unless they are passed. A local variable that needs to be used in multiple functions should be passed to the necessary functions.

### An argument is any piece of data that is passed into a function when the function is called. A parameter is a variable that receives an argument that is passed into a function. A global variable can be accessed by any function within the program but should be avoided if possible.

## Problem Description

A retail company must file a monthly sales tax report listing the total sales for the month and the amount of state and county sales tax collected. The state sales tax rate is 4 percent, and the county sales tax rate is 2 percent. Write a program that asks the user to enter the total sales for the month. The application should calculate and display the following:

* The amount of county sales tax
* The amount of state sales tax
* The total sales tax (county plus state)

## Problem Algorithm

1. Get the total sales for the month.
2. Multiply the total sales by .04 to calculate the state sales tax.
3. Multiply the total sales by .02 to calculate the county sales tax.
4. Add the state tax and county tax to calculate the total sales tax.
5. Display the calculated county tax, state tax, and total sales tax.

## Solutions

1. This program is most easily solved using just four variables. Declare the variables that you will need in the program, using the proper data type and documenting the purpose.

|  |  |
| --- | --- |
| **Variable Name** | **Purpose** |
| Declare Real totalSales | Stores total sales the user inputs |
| Declare Real stateTax | Stores calculated state sales tax |
| Declare Real countyTax | Stores calculated county sales tax |
| Declare Real totalTax | Stores calculated total tax |

1. Given the problem for this program, what modules might you consider including? Provide additional module names and describe the purpose of each module.

|  |  |
| --- | --- |
| **Module Name** | **Purpose** |
| Module inputData() | Allows the user to enter required user input |
| Module calcState() | Calculates the state sales tax |
| Module calcCounty() | Calculates the county sales tax |
| Module calcTotal() | Calculates the total tax |
| Module printTotals() | Displays the state sales tax, county sales tax, and total sales tax |

1. Write the pseudocode. Also, when writing your modules and making calls, be sure to pass necessary variables as arguments and accept them as reference parameters if they need to be modified in the module.

// Pseudocode goes here

Module main ()

Call inputData ()

Call calcState (totalSales)

Call calcCounty (totalSales)

Call calcTotal (stateTax, countyTax)

Call printTotals (stateTax)

Call printTotals (countyTax)

Call printTotals (totalTax)

End Module

Module inputData ()

Declare Real totalSales

Display “Input total sales: “

Input totalSales

Return totalSales

End Module

Module calcState (Real totalSales)

Declare Real stateTax

Set stateTax = totalSales \* .04

Return stateTax

End Module

Module calcCounty (Real totalSales)

Declare Real countyTax

Set countyTax = totalSales \* .02

Return countyTax

End Module

Module calcTotal (Real stateTax, Real countyTax)

Declare Real totalTax

Set totalTax = stateTax + countyTax

Return totalTax

End module

Module printTotals (Real stateTax)

Display “The state tax is “, stateTax

Module printTotals (Real countyTax)

Display “The county tax is” countyTax

Module printTotals (Real totalTax)

Display “The total tax is “, total tax

1. Translate your pseudocode from the previous section to actual code using C++. Read the following program prior to completing the lab. **Call your C++ file, “monthly\_salesport.cpp”.**

Consider the following C++ functions for your program:

* main that calls your other functions
* inputData that will ask for the monthly sales
* calcCounty that will calculate the county tax
* calcState that will calculate the state tax
* calcTotal that will calculate the total tax
* printData that will display the county tax, the state tax, and the total tax

If your program is correct, its output should look as follows (you might get fewer than 2, or more than 2 digits after the decimal point; this is okay):

Welcome to the total tax calculator program

Enter the total sales for the month: 12567<Enter>

The county tax is 251.34

The state tax is 502.68

The total tax is 754.02

When your code is complete and runs properly, screenshot the output.  Be sure to show both the source code and the output. You may not be able to show all of the source code but you should show all of the output.

Describe your experience and lessons learned with this lab.  What is the most challenging aspect of this lab?

This lab is relatively simply if you use the tools we learned in the lecture. One thing that can definitely be tricky is remembering what you are returning from your modules, and where your variables are declared. Not knowing this can lead to issues with scope and parameter names. The most challenging aspect of this lab for me was probably the pseudocode. I knew the steps needed to achieve the results, but I definitely struggled when it came to translating my ideas into pseudocode.

# What to Submit

Submit the following documents to the module named "Topic 3: Lab 3a" using the "Lab 3a Activity" assignment:

* This document completed and saved.
* The source code file called, “monthly\_sales\_report.cpp” that was created for Activity #4.
* A screenshot of the execution of your program. **Include the entire window of the editor in which you wrote your code and executed it**. Be sure not to include other things in the image like your desktop.