**Lab 2: Modules**

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

Name: **Ulric Aird**

# Lab 2.1 – Algorithms

This lab requires you to think about the steps that take place in a program by writing algorithms. Read the following program prior to completing the lab.

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)

**Step 1:** Examine the following 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.

**Step 2:** Given a total sales of $27,097, answer the following:

What is the calculated state tax? **$1,083.88**

What is the calculated county tax? **$541.94**

What is the calculated total tax? **$1,625.82**

# Lab 2.2 – Pseudocode and Modules

|  |
| --- |
| Critical Review    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 name() *Statement*  *Statement Etc.* End Module    Calling a module is normally done from the main() module such as:    Call name()    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()  Real Integer 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)  Number = 20  End Module    // number does not to be sent as a reference because  // number is not going to be modified Module printData(number)  Display "The number is ", number  End Module |

This lab requires you to think about the steps that take place in a program by writing pseudocode. Read the following program prior to completing the lab.

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)

**Step 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 countyTax | Stores county tax amount |
| Declare Real stateTax | Stores state tax amount |
| Declare Real totalTax | Stores total amount of tax when added together. |

**Step 2:** Given the major task involved in this program, what modules might you consider including? Describe the purpose of each module. (Reference: Defining and Calling a Module, page 106).

|  |  |
| --- | --- |
| **Module Name** | **Purpose** |
| Module inputData () | Allows the user to enter required user input |
| Module calcCounty() | Calculates the input by 0.02 to get the county tax |
| Module calcState() | Calculates the input by 0.04 to get the state tax. |
| Module calcTotalTax() | Adds up the county and state tax to get the total amount of tax |
| Module main() | This is where the main function of the code goes |
| Module printData() | This is where you display to county, state and total tax. |

**Step 3:** Complete the pseudocode by writing the missing lines. (Reference: Defining and Calling a Module, page 106). 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. (Reference: Passing Arguments by Value and by Reference, page 127).

Module main ()

// Declare local variables

Declare Real totalSales

Declare Real countyTax

Declare Real stateTax

Declare Real totalTax

// Function calls

Call inputData(totalSales)

Call calcCounty(totalSales, countyTax)

Call calcState(totalSales, stateTax)

Call calcTotalTax(totalTax, stateTax, countyTax)

Call printData(totalTax, stateTax, countyTax)

End Module

// this module takes in the required user input

Module inputData(Real Ref totalSales)

Display "Enter the total sales for the month."

Input totalSales

End Module

// this module calculates county tax

// totalSales can be a value parameter because it is not

// changed in the module.

// countyTax must be a reference parameter because it is

// changed in the module

Module calcCounty(Real totalSales, Real Ref countyTax)

countyTax = totalSales \* .02

End Module

// this module calculates state tax

Module calcStateTax(Real totalSales, Real Ref stateTax)

stateTax = totalSales \* 0.04

End Module

// this module calculates total tax

Module calcTotalTax(Real Ref totalTax, Real stateTax, Real countyTax)

totalTax = stateTax + countyTax

End Module

// this module prints the total, county, and state tax

Module printData(Real totalTax, Real countyTax, Real stateTax)

Display “County Tax: “, countyTax

Display “State Tax: “, stateTax

Display "Total Tax: “, totalTax

End Module