1. Problem Definition

The problem to be solved is finding the amount paid to State Taxes, County taxes as well as the total amount paid for a purchase made from a given amount.

1. Analysis

I gave the inputted value to be stored by the program as variable “purchase” for reference by other calculations, holding a floating point number entered by the user as the amount the purchase cost as prompted. As for output variables “StateTax” holds the calculation for “purchase \* 0.05” to get the amount to be paid in state taxes and holds it. “CountyTax” holds the calculation and result of “purchase \* 0.025” and lastly “Total” holds the amount of variables purchase, StateTax, and County tax added together to come to the total cost (purchase + StateTax + CountyTax)

1. Design

Ask Purchase amount

Store as value “purchase”

Multiply “purchase” by State Tax of “0.05” 5%

Store result as “StateTax”

Multiply “purchase” by county tax of “0.025” 2.5%

Store result as “CountyTax”

Add values CountyTax, StateTax and purchase

Store sum as “Total”

Display all stored values with title in 4x2 grid-like

1. Implementation

I programmed this on a PC running Windows 7 in the IDLE environment. The program preforms as expected when numerals are input, using random numbers and checking answers, numbers in final results will round down if floating point numbers beyond 2 decimal places, keeping them as is through calculations but rounding to two decimal places when printing the final results (tested by checking the values within the program, printing the assigned variable.) numbers with less than two decimal digits preforms as expected. Non-numeral inputs will crash the program as it is not able to convert any random string into numbers and cannot continue its processes as it will not be able to turn it into a floating point or integer number. Formatting holds up until numbers approach the 100-trillions, all amounts below that place (up to 99,999,999,999,999.99) as an output keep numbers aligned properly along the decimal points in each.