**ITCS 1212L – Postlab 5**

**It is recommended that you write your program in codeblocks and then copy and paste the code and submit it.**

1. (20 Points) Write a function named times\_N. The function should gets 2 integer parameters named number and N. When times\_N is called, it should display the product of number times N. Make sure to test this function with different numbers in the main().

#include <iostream>

using namespace std;

void timesN (int, int); //This is my prototype before main() begins. It belongs with the include statements and using namespace etc…

//I am declaring N as a global constant variable before main()

int N = 25;

//This section is main()

main()

{

int number1(5), number2(10);

timesN(number1, N);

timesN(number2, N);

return 0;

}

//This next section is my function itself. It follows directly after main().

void timesN (int number, int N)

{

cout << endl << (number \* N) << endl;

}

//The output for my code is simple. It displays the number 125, then goes down 2 lines and displays the number 250. The test worked!

1. (20 Points) Write a new function named times\_N2. Again, this function also gets 2 integer parameters named number and N. When times\_N2 is called, it should display the product of number times N. However, N will be doubled upon return. Make sure to test this function with different numbers in the main().

//For this problem, I am going to declare N as a local variable within main(). Here’s the prototype.

#include <iostream>

using namespace std;

void timesN\_2 (int, int&);

//Here is main again

main()

{

int number1(5), number2(10), n(1);

timesN(number1, N);

timesN(number2, N);

return 0;

}

//Now for the function

void timesN\_2 (int number, int& n)

{

cout << endl << (number \* n) << endl;

n \*= 2;

}

//Now the output is 125, down 2 lines, then 500.

1. (20 Points) Write a function called compute which takes in three parameters- an integer quantity, a double price, and a double for salesTax, and returns the totalSales (product of quantity and price), totalTax (sales tax assigned to totalSales), and total as the sum of totalTax and salesTax. Make sure to include the main() that tests this function. //This problem is asking us to return 3 values from a function, which is impossible without creating and return structures (which I do not entirely understand how to do yet). There are several different ways that could make this program function as desired, but none of them will pass the 3 original variables and return 3 new variables to main. One possible way is to use reference points to change values within the function by going to the actual memory address of some of the original variables, but since the instructions were much more clear in the beginning that I was only passing three (int, double, double), I decided to go with a different approach. I could’ve additional functions to deal with sales tax and the other returning values, but I didn’t want to clog this postLab anymore than I have with this very long paragraph. I finally decided on following the instructions to the point where it says to return totalSales, and then perform the other actions within main() itself. If this question was asking for a different, more specific method, please get back to me and give me a chance to fix this as I am unsure what else this problem could be asking.

//Before main()

#include <iostream>

using namespace std;

double compute(int, double, double);

main()

{  
 int quantity(10);

double price(2.25), salesTax(.08), totalSales(0), totalTax(0), total(0);

//This satisfies passing the value of 3 variables and returning totalSales

totalSales = cashRegister(quantity, price, salesTax);

//The instructions then ask for totalTax (sales tax assigned to totalSales). This

means that I would replace the value of totalSales with the value of salesTax because I

would be “assigning” salesTax to totalSales. I am not going to do this of course, because

that would be nonsensical and delete our totalSales value. My best guess is it wants me

to assign totalTax with the value of (totalSales \* salesTax).

totalTax = totalSales \* salesTax;

//Again, the instructions are asking me to do something very silly. I am being told to assign the variable ‘total’ with the sum of totalTax and salesTax. This would be adding the actual tax amount to the tax percent (.08), which means absolutely nothing when trying to figure out the total value of the item we are selling. What I think it wants me to do is take the total of (quantity and price) and multiply it by the tax percent (.08), then add that value to the total again (quantity and price), to give the final amount. This step was unnecessary because I had simply added 1.0 to the salesTax value when multiplying in the function. (quantity \* price) \* (1 + salesTax) IS the final answer. Final sale after tax.

//This problem didn’t ask to display anything, so I’m just going to reiterate the logic behind getting different values the user might be interested in.

//Cost before tax: (quantity \* price)

//Cost in sales tax: (quantity \* price \* .08/salesTax)

//Cost after tax is applied: (quantity \* price) \* (salesTax + 1)

return 0;

}

double compute(int quantity, double price, double salesTax)

{

double totalSales(0);

totalSales = (quantity \* price) \* (1 + salesTax);

return totalSales;

}

1. a) (10 Points) What is the output of the following program:

#include <iostream>

using namespace std;

void myFunc1(int var) {

var = 50;

cout << var << endl;

}

void myFunc2(int &var) {

var = 50;

cout << var << endl;

}

int main() {

int var = 100;

myFunc1(var);

cout << var << endl;

myFunc2(var);

cout << var << endl;

return 0;

}

50

100

50

50

b) (10 points) Add a new function void myFunc3(int m, int &n) which gets two variables m and n and will return the sum of the two numbers. Make sure to test it in main().

void myFunc3(int m, int& n)

{

int total;

total = m + n;

cout << total << endl;

}

int main()

{

int m(2), n(3), total(0);

total = myFunc3(m, n);

cout << total << endl;

cout << “Both values are the same despite being displayed from different functions”;

return 0;

}

5. (20 Points, 5 points each)

* 1. Values that are sent into a function are called [This was worded a little strangely, so I’m going to expand to clarify, to ensure I answer this correctly]. When a function is called, it passes its arguments to the function so that it can receive its parameters.
  2. If a function doesn’t return a value, the word VOID will appear as its return type.
  3. **True / False**: Functions should be given names that reflect their purpose.

**True**

* 1. **True / False**: Function prototypes are terminated with a semicolon.

**True (assuming you’re not running the function before main and declaring the prototype at the same time you write its execution)**