

In this assignment you will implement a program called *'nursery_inv3.cpp'* to process the customer purchase orders for a nursery. The purchase orders will be stored in order records. Each order record contains nine fields, which are as follows: 1) a plant name (string, no spaces), 2) a county name (string, no space), 3) the cost of the plant (double), 4) the quantity of plants in the purchase (integer), 5) the tax on the purchase (double), 6) the net cost of the purchase (double), 7) the tax rate, 8) the discount on the purchase (double), and 9) the total cost of the purchase. Your program will have 3 functions: input, process, and output. Your main program will call (invoke) each function until the end of the data file has been reached. Following are the descriptions of the functionality of each function:

1. The void function **“input”** will have two parameters: an input file stream called **“in”**, and a customer purchase order record called **“order_record”**. Both arguments will be passed by reference. The function will read a plant name (*pname*), the name of the county the plant was purchase in (*cname*), the cost of the plant (*plant_cost*), and the quantity of the plant purchase (*quantity*) into an order record from the data file, *“nursery_stock.txt”*.
2. The void function **“process”** will have one argument, the order record, which will be passed by reference. The function will calculate the *net cost* of the purchase (*net_cost*), the *tax* on the purchase (*purchase_tax*), the *discount* on the purchase (*discount*), and the *total cost* of the purchase (*total_cost*). Please consider the following information to help you implement the necessary calculations:
 - a. The *tax rate (in percent) on a purchase* is based on the *county* where the purchase was made. If the county was *dade*, the tax rate is 6.5%; if the county is *broward*, the tax rate is 6%; if the county was *palm*, the tax rate is 7%.
 - b. The *net cost of a purchase* is calculated by the following formula:
 - $net_cost = (quantity \times plant_cost)$
 - c. The *discount* is based on the *quantity of plants* in the purchase. *The discount is determined as follows:*
 - If *quantity* equals 0, then the *discount* is 0% of the net cost;
 - If $1 \leq quantity \leq 5$ then *discount* = 1% of the net cost; $6 \leq quantity \leq 11$ then *discount* = 3% of the net cost; if $12 \leq quantity \leq 20$ then *discount* = 5% of the net cost; $21 \leq quantity \leq 50$ then *discount* = 8% of the net cost; *quantity* > 50 then *discount* = 12% of the net cost). **Apply discount after the net cost has been calculated.**
 - d. The *tax on a purchase* is calculated by the following formula:
 - $purchase_tax = (net_cost \times tax_rate / 100)$ **(drop/100 if you converted the rate from a percentage)**
 - e. The *total cost of a purchase (rounded to the nearest hundredth)* is calculated by the following formula:
 - $total_cost = net_cost + purchase_tax - discount$.

Note: **All tax and cost calculations should be rounded to the nearest hundredths.**

3. The void function “**output**” will have two arguments, an output file stream and an order record. The output file stream will be passed by reference, and the order record will be passed by value using the modifier “const”. Call the output file stream, **out**, and the order record, **customer_order**. The function will print every field of an order record to the file, “***sells_results.txt***”. The fields should be printed in the following order: plant name, county name, plant cost, quantity of plants, net cost of purchase, tax rate, purchase tax, discount, and total cost. The following table shows the format.

Use the following format information to print the variables:

Field	Format	Width	Justification
Plant Name	string	15	left
County Name	string	10	left
Plant Cost	XXXX.XX	7	right
Quantity of Plants	XXXXXX	6	right
Net Cost of Purchase	XXXXX.XX	8	right
Tax Rate	XX.XXX	6	right
Purchase Tax	XXXX.XX	7	right
Discount on Purchase	XXXX.XX	7	right
Total Cost of Purchase	XXXXXX.XX	9	right

You may implement more functions if you find it necessary. Please start the assignment ASAP, and ask questions to make sure you understand what you must do. **It is always good to start with the skeleton program I provide.** Remember to follow all style rules and to include all necessary documentation (consistent, indentation, proper variable names, pre/post conditions, program header, function headers, and so forth).

See the sections below called “**Input Stream**” and “**Output File Format**” for more information.

Input Stream:

In the assignment you will declare one ifstream to bind your input to the file “*nursery_stock.txt*” to an input file stream. Whenever a program performs file i/o you must include the “*fstream*” library. **Add the following statements to your program:**

For source file, “*nursery_inv3.cpp*”

- Add “#include<iostream>” to your #include statements in your source file.
- Add “#include<fstream>” to your #include statements in your source file.
- Add “#include<string>” to your #include statement in your source file.
- Add “#include<iomanip>” all formatting of output

Format of the input data file (input filename is “nursery_stock.txt”):

Do not include column titles in your data file. Field as order as follows: plant name, county name, plant cost, quantity.

Owl	dade	10.55	100
Hibiscus	broward	15.82	15
Rose	dade	9.99	45
carnation	palm	7.99	32
Rose	palm	7.99	60
Widow	palm	25.75	5
carnation	dade	12.55	10
carnation	dade	12.55	8
Lilly	broward	6.92	150
xerabtgemum	palm	13.63	50
Yarrow	dade	22.85	20
Zenobia	palm	37.19	32
zephyranthes	broward	62.82	40
Daisy	broward	15.99	80
aconitum	dade	30.02	72
amaryllis	dade	16.14	65
Begonia	broward	18.45	3
Bellflow	broward	2.96	200
bergenia	palm	85.92	10

Output File Format:

Consider the following sample output table when designing and implementing the function “output”:

The output is in the following order: plant name, county name, plant cost, quantity of plants, net cost of purchase, tax rate, purchase tax, discount, and total cost.

owl	dade	10.55	100	1055	0.065	68.58	126.6	996.98
hibiscus	broward	15.82	15	237.3	0.06	14.24	11.87	239.67
rose	dade	9.99	45	449.55	0.065	29.22	35.96	442.81

Using the information in the data file “*nursery_stock.txt*” you should produce and print the following output to the file, “*sells_results.txt*”. Your output should not contain any titles, but the output must be in the proper order as stated in the assignment. The following contains an example of what your program should produce and print to the file, “*sells_results.txt*. You should double check the results.

owl	dade	10.55	100	1055.00	0.065	68.58	126.6	996.98
hibiscus	broward	15.82	15	237.30	0.060	14.24	11.87	239.67
rose	dade	9.99	45	449.55	0.065	29.22	35.96	442.81
carnation	palm	7.99	32	255.68	0.070	17.9	20.45	253.12
rose	palm	7.99	60	479.40	0.070	33.56	57.53	455.43
widow	palm	25.75	5	128.75	0.070	9.01	1.29	136.47
carnation	dade	12.55	10	125.50	0.065	8.16	3.76	129.89
carnation	dade	12.55	8	100.40	0.065	6.53	3.01	103.91
lilly	broward	6.92	150	1038.00	0.060	62.28	124.56	975.72
xerabtgemum	palm	13.63	50	681.50	0.070	47.71	54.52	674.69
yarrow	dade	22.85	20	457.00	0.065	29.71	22.85	463.85
zenobia	palm	37.19	32	1190.08	0.070	83.31	95.21	1178.18
zephyranthes	broward	62.82	40	2512.80	0.060	150.77	201.02	2462.54
daisy	broward	15.99	80	1279.20	0.060	76.75	153.5	1202.45
aconitum	dade	30.02	72	2161.44	0.065	140.49	259.37	2042.56
amaryllis	dade	16.14	65	1049.10	0.065	68.19	125.89	991.4
bogonia	broward	18.45	3	55.35	0.060	3.32	0.55	58.12
bellflow	broward	2.96	200	592.00	0.060	35.52	71.04	556.48
bergenia	palm	85.92	10	859.20	0.070	60.14	25.78	893.57

Use the following skeleton to help you start this program which is included as a cpp file in the module: nursery_inv3.cpp

```
//Your program header will go here

//Include the appropriate libraries

// add using namespace std;

//Put the declaration for order record here

//Prototypes for your functions: input, output, and process will go here

//Function Implementations will go here with their headers

//Your driver (main program) to test the program will go here
int main()
{
    //declare an order_record object
    //declare an input file stream
    //declare an output file stream
    //open the input file stream
    //open the output file stream

    //check to see if the input file stream opened okay

    //if input file stream opened okay then
    // read until end of file using a while loop
    //   inside the loop call input, process, output

    //close input file stream
    //close output file stream
    return 0;
}
```