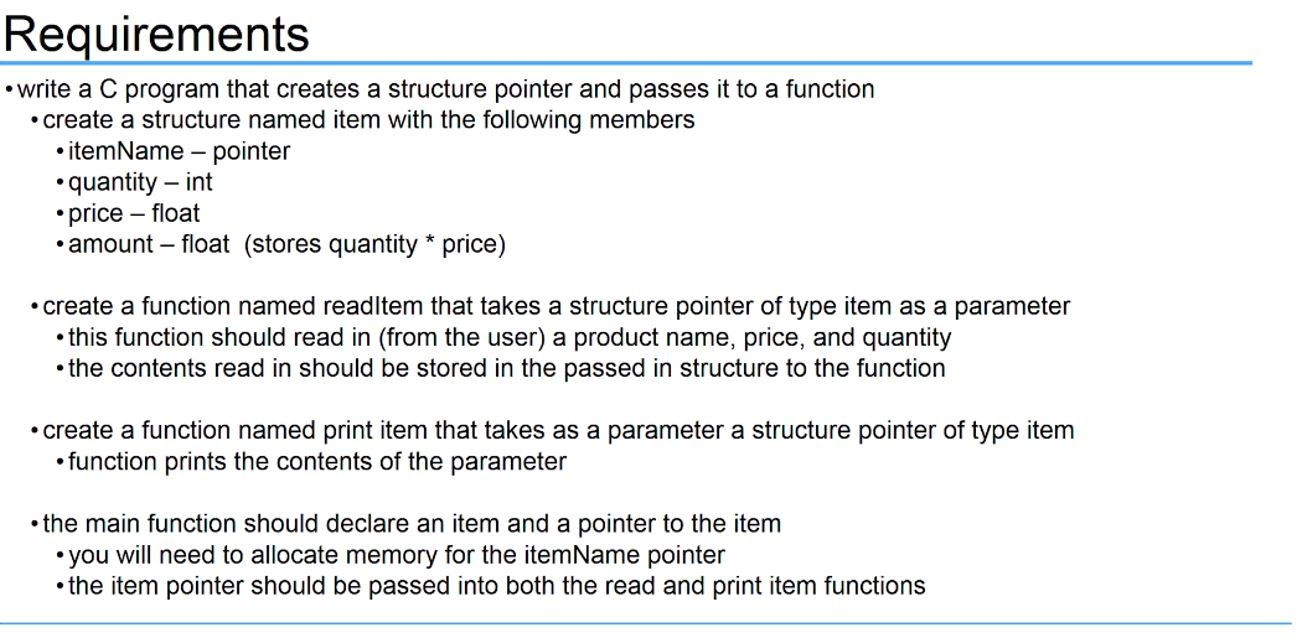
**STRUCTURES & POINTERS**

Lab Exercise # 8

Submitted by:**Sumalinog, Ethan Chav Isaac P.**

Submitted to:**Mr. Nasvin Del Rosario**

**Challenge:**



**Solution:**

I first created a *structure item* data type with the 4 members: *char \* itemName*(a pointer)*, float price, int quantity* and *float amount*(the price times the quantity).

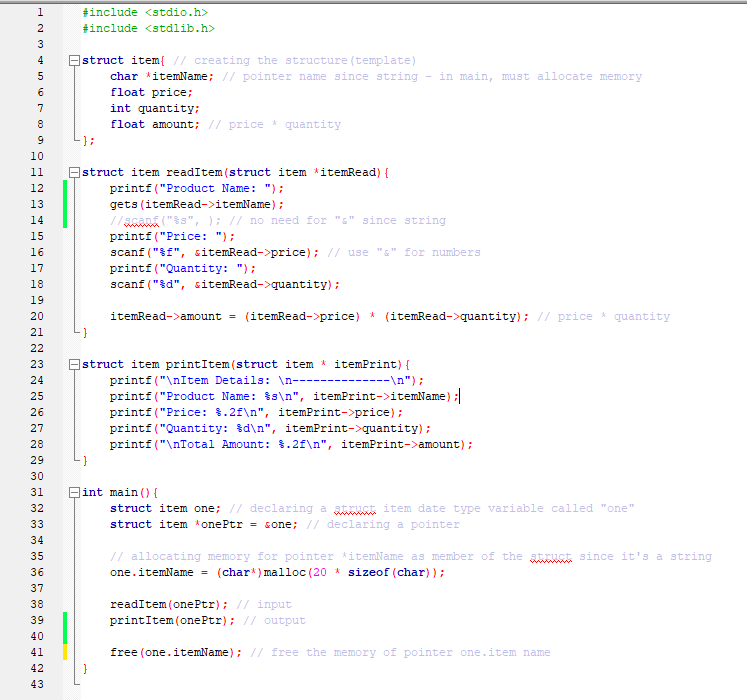
I then created a function with data type *struct item* with the name *readItem* that takes a parameter of data type *struct item* pointer named *\*itemRead*. Here, this gets input from the user, which are *itemName*(Product Name)*, price* and *quantity.* For the amount, it’s just the price multiplied by the quantity, stored in amount.These are all placed in the \*itemRead pointer to its respective member of the *struct item* data type.

Afterwards, I created another function that prints out what was inputted by the user. It’s called *printItem* of the data type  *struct item* that takes in the parameter of *struct item* pointer *\*itemPrint*. This just prints out the *itemName, price, quantity* and *amount*.

In the *int main(),* I first declared an instance of *structure item* *one* and made a pointer *\*onePtr* pointing to the address of *one.* For the *itemName*, I allocated memory of 20. I called out the function *readItem* with the argument of pointer *onePtr* for the input and then the function *print*Item with the same pointer to print out the contents. I then used *free(one.itemName)* to free the memory allocated for *one.itemName*.

What we’re doing here is that we are using pointers to pass out values using pass by reference. We do not put the entire structure in the parameters of the functions but only the what is to be inputted by the user. By making a struct pointer, we are only using the addresses of the pointer. In this example, from the *struct item one*, we made a pointer *struct item* \* *onePtr* pointing to the address of *one.* We are then only using the *onePtr* to let it be used in the parameters of functions.

**Code:**



**Output:**

