CS 2401 – Spring 2023

Labs 9 & 10 – Inheritance and Virtual Functions

UML diagram and Class implementations Due: 11:59 PM Friday, March 31st

Main program Due: 11:59 PM Friday, April 7th

Congratulations, you are opening your own business! Luckily, you are a great programmer, so you are designing the order management system for your business. You will be offering 5 different products to your customers, however, the attributes that would be used to describe those things can vary greatly. You would not use the same variables (or attributes) to describe a sweater that you would use to describe a pair of pants, let alone a soccer ball or the fancy new soaps you make.

For this project you are to create a group of classes that describe the products that your business will offer. Since they have the common feature of being sold by you, there will be a parent class called something like "Product" and then a minimum of five other classes which are children of this class. We're not going to be doing any real calculations using these items, so all they really need are input and output functions, which will need to be virtual in the parent.

You are also to construct, for 10% of your grade, a UML class diagram which describes your classes, but does not need to describe the application. This diagram is to be constructed using Star UML and should be in the form of a .mdj file, that is included with your project submission. (Star UML is available for free at https://staruml.io/download)

The application for this project will declare an STL list of pointers of the parent class (Product) which will hold dynamically allocated products that have been ordered by your customers. The user will be presented with a menu from which they can choose a product that they would like to order. They will then be taken to an input function for that item where they can enter its attributes. Once the customized product is constructed, it is put into the list with the push_back member function.

The menu should also offer a means to view all the items in the list. Since the list class has no "show_all" function, we will have to rely on the iterator to do this, and note that since we are storing pointers to items instead of the items themselves, and since non-member overloaded operators don't work well with this paradigm, the syntax for outputting each thing will look like: `(*it)->output(cout);`

Finally, you need to write a backup file system so that a user does not have to reenter all the products that have been ordered every time they start the program. This backup file is to be invisible to the user (call it orders.txt). The program simply looks for the file, and if it is not found it is assumed that this is the first use of the program, but if it is found the product orders are put into the list the same way as they were when entered from the keyboard. Following the loading of the list from a file the user is taken to the menu where they can see the orders that are waiting to be filled or add new orders to the list. You do not have to implement an option to remove things from the order list (but you can if you want to).

Please note: You are NOT writing a container but are instead using the STL list class. (Available if you #include <list>.) For this project you will just need the push_back function and the iterator, which is bi-directional, but which you will only be using in a forward direction for order output.

Also note, I am giving you no code for this project – you are to write it all yourself, and variety and creativity of the child classes is a factor in determining your grade. You are to write a total of six classes and the main which will include the menu that works for your distinct set of classes.