**Supermarket**

For this assignment you‟ll write a simple class hierarchy and use functions to carry out common tasks across all child classes.

**The assignment problem:**

You are asked to develop an application to help a Supermarket keep track of their products (food and electronics). Your application will read input from two input files (Food.txt and Electronics.txt), and output a file “Output.txt” that will have a summary for products that will have a discount on them or products that have warranty.

To implement this application, you will need to have 3 classes:

1. Product class (Base Class)
2. Food class
3. Electronics class

There are some common features among all products such as barcode, name, price, tax and discount. Also, there are still some features that are different.

For the implementation of your application you need to have the following features:

* 1. **Base class ‘Product class’:**
  + Constructor that takes parameters for the barcode and price, name, tax and discount. The barcode must be bigger than 0, otherwise the barcode must be set to 0 - Getters and Setters for barcode, price, name and others.
  + A method called update(). This function will have an argument of type ofstream pass by reference. This function will not return a value. It will output data to a file.

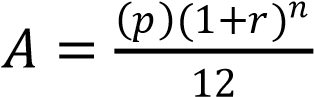
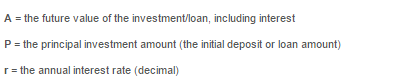
* 1. **Food Class:**
  + Will inherit from Product class everything.
  + Add two more member variables: date\_produced, number\_of\_days.
  + Constructor.
  + Getters and Setters.
  + A method called ReadData(). It takes an istream passed by reference as its only parameter. It reads in an integer (barcode), name, price, date\_produced, number\_of\_days (in this order). This method returns a boolean value: true if all data was read successfully, false otherwise. Use this member function to read data from the first file (“Food.txt”)
  + Redefine update() function. This function will check number\_of \_days the item sat on shelf. o If number\_of\_days less than 10, price will remain the same o If number\_of\_days between 10 -14, item will have a discount of 20% o If number\_of\_days between 15-30, item will have a discount of 50% o Otherwise item will have a discount of 80%.

o This function will output to “Output.txt” the price after discount.

**3- Electronics class:**

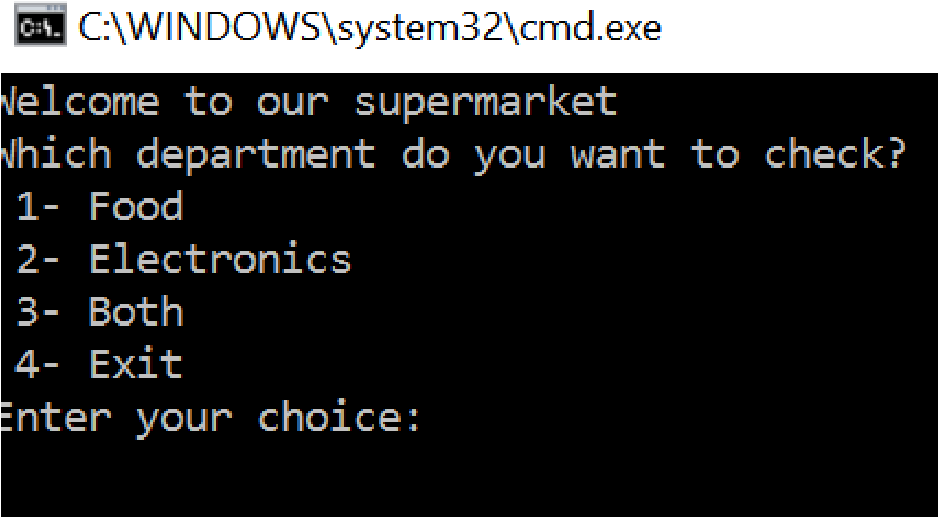
* This class will inherit everything in Product class. - Will add one more member variable „warranty‟ of type int
* Constructor
* Getters and Setters
* A method called ReadData(). It takes an istream passed by reference as its only parameter. It reads in an integer (barcode), name, and price. This method returns a boolean value: true if all data was read successfully, false otherwise. Use this member function to read data from the second file

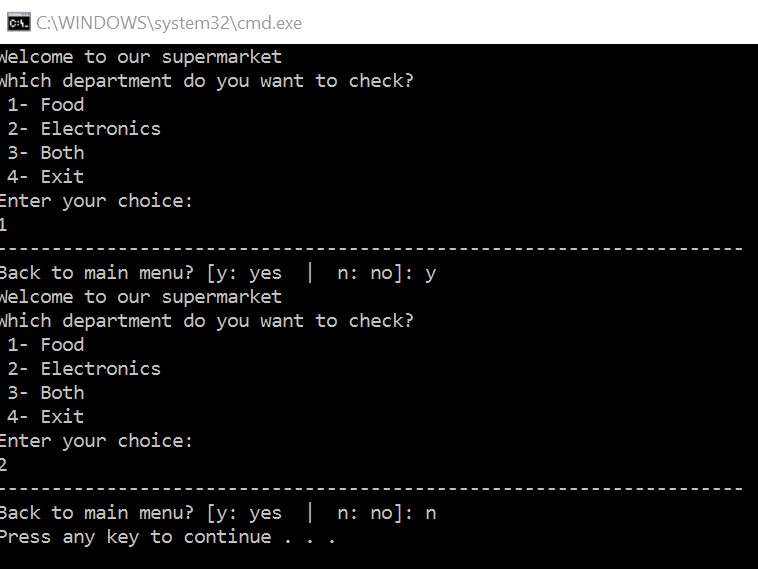
(“Electronics.txt”)

* A 9% tax will be added to all electronics
* Redefine update() function. This function will check the price of item and select one of the following options: o If the price of the item is less than 200, then the item will have warranty for 1 year and there is no financing option available
  + If the price is 200-1000, then the item will have a warranty for 3 years, and it can either be bought in cash or it has a financing option to be paid over 12 months with a 1% compound annual interest of the price after tax.
  + Otherwise, the item will have 5 years warranty and it can either be bought in cash or it has a financing option to be paid over 24 months with a 1% compound annual interest of the price after tax. o  Where n is number of months/12.
* 
  + This function will output to “Output.txt” years of warranty, total price after adding taxes on items and options of payment.

**4- Main function:**

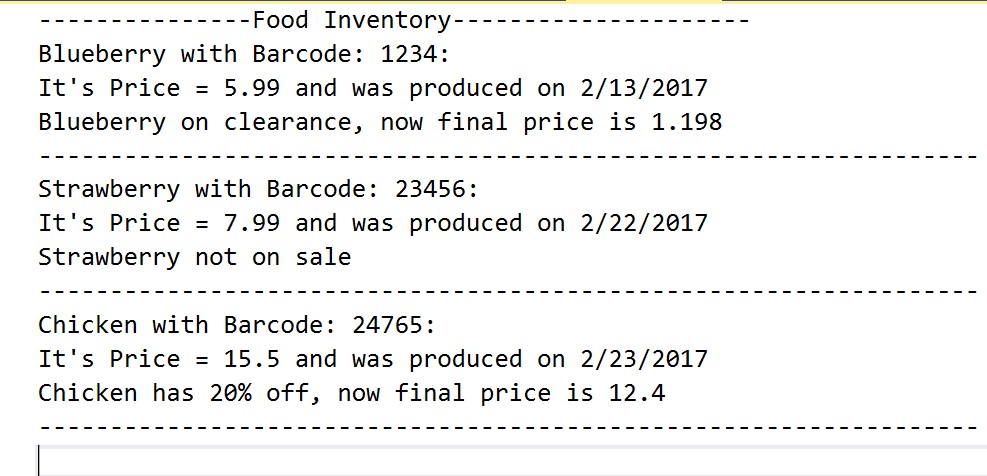
* Declare two objects: one for Food class and the other for Electronics class. - Your application starts by providing the following :



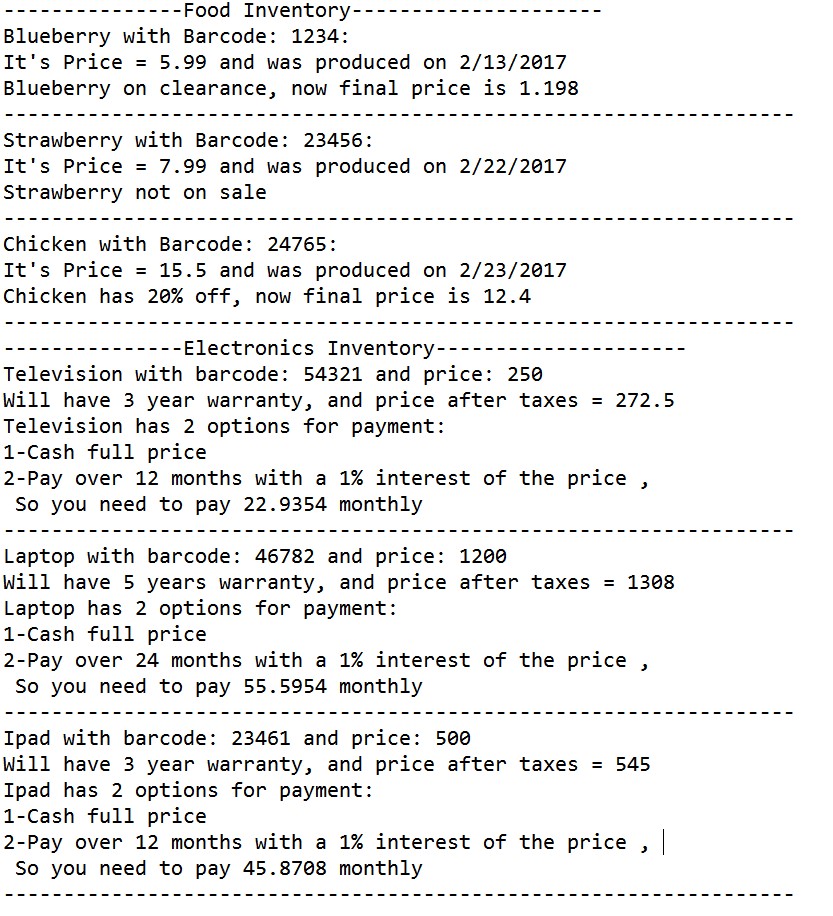


* As you can see, after user input his choice, the output will not be printed on the screen. It will be printed to the output file „Output.txt‟.
* After user finish his/her transaction, he/she will be asked if he/she wants to return to main menu again.

* Sample of the „output.txt‟, if user chose “1”:



* Sample of “output.txt” , if the user chose‟3‟



**Submission:**

* Your application must be implemented as an OOP (uses a class, header file, and a main program). Otherwise, it will receive no score.
* Zip up your entire project folder and submit the zip file to Blackboard by the deadline.