**Structure**

Main:

* Greet user with the name of the store and explain what the program does.
* Call the menu interface
  + Display menu()

Class (Products):

* Have a list for the products offered
  + Create private instance variables: number, name, price
    - These will belong to each object created of the class
    - Create getters and setter functions for the private variables
  + Should include name of product and price.
  + Product\_list of objects, should display its number, name, price.
    - 1. Hotdog 2.00
    - 2. Pizza 3.00
    - Etc…
* Make a constructor named after the Store
  + ( ) parameter should accept an int, string, and price
    - number - the products chosen by the user
    - name – this will provide the name of product
    - price – price of the item
    - Call the next method to display the list for user.
* Create an array that collects the user’s input.
* Create a method that cycles through the collection of user inputs
  + It will print every item selected by the user.
  + Should not return anything, make it void.
  + Make it static so it can be used elsewhere without object

Interface (Menu):

* Create a menu function that prints out the menu when called.
  + This will be the entire program
  + Create a do while loop that records every integer input from the user.
    - If the input is within range, then it will be collected and added to their list.
    - Else if, they will be greeted with an error and be given a reentry.
    - Else if the input is 0, then exit loop and onto next step.
  + Call to make a Receipt of the items
    - Display the items they are purchasing along with their total cost of purchase.
  + Let them pick a delivery date.

Methods:

* (PrintProduct) make a method that shows the item selected
  + Method should not return a value
  + Pass in the object that you want to be shown.
  + Display the item and the total cost after each use.
* (collectItem) make a method that collects each item selection from user.
  + Collects user choice and passes it into a switch statement to be printed
  + Prints that it has added the item and displays item information.
* (Delivery) create a method that collects the delivery date
  + Displays today’s date
  + lets user create a delivery date
  + displays delivery date entered
* (Receipt) Prints user Receipt
  + Displays each item that was selected for purchase
  + Displays total purchase cost.
  + Calls Deliverymethod
* (PrintArray) cycles through the array by its size and prints each value
  + Uses a for loop to display each item from array
* (PrintItemArray) Prints requested value in array