Assignment 2

Hot Dogs	40 pts.
Improvement of the InventoryItem Class	20 pts.
Student Class	40 pts.
TOTAL	**100 pts.**

Part 1

Hot Dogs

Start with the file **HotDog.cpp.**

You operate several hot dog stands distributed throughout town. Define a class named `HotDogStand` that has a member variable for the hot dog stand's ID number and a member variable for how many hot dogs the stand has sold that day. Create a constructor that allows a user of the class to initialize both values.

Also create a method named `JustSold` that increments the number of hot dogs the stand has sold by one. The idea is that this method will be invoked each time the stand sells a hot dog so that we can track the total number of hot dogs sold by the stand. Add another method that returns the number of hot dogs sold.

Finally, add a static variable that tracks the total number of hotdogs sold by all hot dog stands and a static method that returns the value in this variable.

Write a main function to test your class with at least three hot dog stands that each sells a variety of hot dogs.

Hint: Recall that static variables must be initialized outside of the class definition.

Part 2

Improvements of the InventoryItem Class

Take the code of the InventoryItem class that was rewritten and debugged (provided in InvItem_FIXED.cpp). Add two member functions to the class:

- Copy constructor

- Overloaded assignment operator. Make sure that your operator will not malfunction when called with this sample code:

```
InventoryItem h ("hammer", 2, 3.0);
h = h;
```

In main() explicitly call copy constructor and assignment operator in order to test them.

Student Class

Start with **Student.cpp** file that contains class definition and function calls in main.

Create a class named `Student` that has three member variables:

`name`	– A string that stores the name of the student
`numClasses`	– An integer that tracks how many courses the student is currently enrolled in
`classList`	– A dynamic array of strings used to store the names of the classes that the student is enrolled in

Write appropriate **constructor(s), mutator, and accessor methods** for the class **along with the following:**

- A method that inputs all values from the user, including the list of class names. This method will have to support input for an arbitrary number of classes.
- A method that outputs the student name and list of all courses.
- A method that resets the number of classes to 0 and the classList to an empty list (NULL).
- An overloaded assignment operator that correctly makes a new copy of the list of courses.
- A destructor that releases all memory that has been allocated.

Write a main function that tests all of your functions.

Hint: Recall that ***cin >> str*** statement leaves a newline character in the buffer. This can be a problem if you are mixing ***cin >> str*** and ***getline***. Use ***cin.ignore()*** to discard the newline.