

Assignment purpose: class member functions, explicit value constructor, friend functions, overload operators, separate compilation.

Update your class implementation (Assignment 5) to include:

Separate the class declaration in a .h file from the member function implementation in a .cpp . See the DayOfYear class code provided with the assignment.

The declaration which includes private member variables, and the prototypes for the member functions and prototypes for the friend functions should be placed in a separate header (.h) file.

The member function implementation will be placed in a (.cpp) file.

The main function (driver) any non member functions (including the friend function implementations) will be placed in a third file: lastname_A8.cpp.

NOTE: Do not use class Vegetable - see the .h file provided with the assignment to help you get started.

Update your class implementation (Assignment 5) to include 2 friend functions and one additional member function.

1. A **friend function** to overload the + operator
 - You decide what should be added together based on your class
 - The vegetable example adds the names of the vegetables together and returns a string,
2. A **friend function** to overload a relational operator (<, <=, ==, >, >=)
3. **Class member function.**

Declare at least 4 objects of your type

1. One object will be initialized with the explicit value constructor
2. One object will be the default object (default constructor)
3. Two objects will be initialized with the default constructor and the new data will be entered by the user through a non member programmer defined function.

Additional instructions:

- Be sure to comment your code
- Be sure to include the class declaration and member function implementation separated and above main.
- Be sure to include function prototypes above main and function definitions below main
- Include a program header COMMENT with the following information:
 - Name, date, course, and a brief description of the assignment
- Test your code in an IDE before submitting
- The file names must match the assignment
- The code must be submitted on time in order to receive credit (11:59 PM on the due date)
 - NOTE: there is a grace period until 8:00 AM the following day
- The assignment allows multiple submissions you may make revisions and resubmit until the final due date
- Late submissions (after 8 AM or sent by email) will not be accepted or graded

Modifying data and submitting it as your own is a fraudulent practice—specifically, plagiarism—and is no different than copying paragraphs of information from a book or journal article and calling it your own (make sure that you work independently and submit only your own work)

This programming assignment is individual work, sharing code is considered cheating,

The use of AI to assist in this assignment is prohibited.

Sample output:

```
Explicit value constructor
// the default constructor was called 3 times
default vegetable is carrot //remain default carrot
default vegetable is carrot //be replaced by user input
default vegetable is carrot //be replaced by user input

-----Getting information from the user-----

Enter veggie name: cauliflower
Enter veggie color: white
Enter the id: 777
Enter cost per pound: 1.67
Enter selling price per pound: 4.12

-----Getting information from the user-----

Enter veggie name: yellow_squash
Enter veggie color: yellow
Enter the id: 1010
Enter cost per pound: 1.11
Enter selling price per pound: 3.48

-----Printing Veggie info-----// Explicit value constructor

ID : 555
Veggie name: onion
Veggie color: white
Cost per pound: $1.50
Selling price per pound: $4.00

-----Printing Veggie info-----//default vegetable

ID : 111
Veggie name: carrot
Veggie color: orange
Cost per pound: $0.75
Selling price per pound: $4.00

-----Printing Veggie info-----

ID : 777
Veggie name: cauliflower
Veggie color: white
Cost per pound: $1.67
Selling price per pound: $4.12
```

-----Printing Veggie info-----

ID : 1010
Veggie name: yellow_squash
Veggie color: yellow
Cost per pound: \$1.11
Selling price per pound: \$3.48

//select 2 objects to demonstrate the friend '+' overload

-----demonstrate the '+' friend function-----

adding 2 veggies: onion and carrot
onioncarrot

//select 2 objects to demonstrate the second friend overload

-----demonstrate the '<' friend function-----

compare the cost per pound of: cauliflower and yellow_squash

comparing the cost per pound of 2 veggies:

yellow_squash cost per pound \$1.11
is less than or equal to cauliflower cost per pound \$1.67

//select one object to demonstrate the added member function

-----demonstrate the added member function-----

How many pounds of onion do you want? 3

The cost per pound is 1.50

The selling price per pound is 4.00

The profit for onion will be: \$7.50

Thank you, have a great day!