

Assignment purpose: C++ AI experiment. There is nothing to submit, the goal is to be able implement similar code by the end of the semester without the aid of an AI tool

- inheritance in C++ (constructors, destructors)
- friend functions
- vector of objects
- file I/O
- class declarations and member function implementation
- programmer defined functions – prototypes and definitions
- loops
- conditions
- C++ basics: input, output, basic math

Objective:

Use AI to design and implement a console-based system to manage a virtual vegetable store using object-oriented programming principles in C++. This project will help reinforce class design, inheritance, file input/output, arrays of objects, and operator overloading.

Program Features:

Your application should simulate a store that sells different vegetables. The system should support the following functionality:

1. **Create a list of vegetables** (using a vector of objects).
2. **Generate unique random IDs** for each vegetable.
3. **Load vegetable data from a file** and save it back on exit.
4. **Support purchasing vegetables** by selecting from a menu.
5. **Use inheritance** to extend the base functionality of a product or item.

Features and Functionality:

1. Inheritance and Class Design:

- The base class `Produce` contains common private attributes for any product (`name` and `color`), provides a default constructor, an explicit value constructor, accessor(getter) and mutator(setter) functions for these attributes.
- The derived class `Vegetable` extends the `Produce` class, adding specific attributes related to vegetables, such as `costPerPound` and `pricePerPound`. The `Vegetable` class also includes a constructor to initialize these attributes and a method to calculate profit based on the number of pounds purchased.

2. Operator Overloading:

- A **friend function** is used to overload the `+` operator, which allows concatenating the names of two `Vegetable` objects (example: "Carrot and Spinach").
- Another **friend function** overloads the `<` operator to compare the `costPerPound` of two vegetables, allowing users to easily compare the prices of different vegetables.

3. File Input:

- The program reads vegetable data from a file (`vegetables.txt`) using `ifstream`. The file contains a list of vegetable names, colors, and their cost and price per pound. The file must be structured as follows:

```
name  
color  
cost per pound  
price per pound
```

This data is used to initialize the `Vegetable` objects in the program.

4. File Output:

- After the user selects a vegetable and provides the number of pounds they wish to purchase, the program calculates the profit and writes the results to a file (`vegetable_results.txt`). The file includes details such as:
 - Vegetable name
 - Price per pound
 - Cost per pound
 - Pounds purchased
 - Calculated profit

5. User Interaction:

- The user is prompted to select a vegetable from a list of available options, then they input how many pounds of the vegetable they want to purchase.
- The program then calculates the profit based on the price per pound and cost per pound and displays the result.
- The program continues running until the user chooses to exit (by selecting 0).

Key Concepts Demonstrated:

- **Inheritance:** The `Vegetable` class inherits from the `Product` class, which allows us to create specialized vegetable objects with additional functionality.
- **Friend Functions:** The program demonstrates the use of friend functions for overloading operators and comparing objects, allowing for cleaner and more intuitive code.
- **File I/O:** The program reads vegetable data from a file to initialize vegetable objects and writes the results of calculations back to a file for persistent storage.
- **Random Number Generation:** To simulate product IDs, the program generates a random number between 1 and 10,000 to assign to each vegetable object.
- **Object-Oriented Design:** The use of classes, constructors, destructors, mutators, accessors, and member functions showcases the object-oriented approach to programming.

Assignment Instructions:

1. Initial Setup:

- Download and set up the provided code in a C++ development environment.
- Ensure that the `vegetables.txt` file exists in the correct directory with the appropriate vegetable data. (do not include white space in the name of the vegetable or color, example Brussel_sprouts)

2. Modify Vegetable Data:

- You may modify the vegetable data in the `vegetables.txt` file to test different scenarios with different vegetables.

3. Run the Program:

- Compile and run the program.
- The program will prompt you to choose a vegetable and enter the number of pounds you wish to purchase.
- After calculating the profit, the results will be written to the `vegetable_results.txt` file.

4. Submit Your Work:

- There is nothing to submit, **the goal is to implement similar code by the end of the semester without the aid of an AI tool**

Grading Criteria:

- **The assignment is not graded**

Conclusion:

This assignment provides practical experience with essential C++ concepts, such as inheritance, file handling, and operator overloading. It also helps develop an understanding of how to manage product data and user interactions in a real-world application.

Additional instructions:

- Be sure to comment your code, all code will be in ONE file
- Be sure to include programmer defined function prototypes above main and definitions below main
- Be sure to include the class declarations (for both the base class and the derived classes) above main
- Member function implementation should be separated from the declarations and above 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,

Sample output (screen):

```
Loaded: Carrot, Orange, $0.3, $0.75
Loaded: Spinach, Green, $0.4, $1.25
Loaded: Brussel_sprouts, Green, $0.5, $1.6
Loaded: Beet, Red, $0.35, $0.9
Loaded: Eggplant, Purple, $0.45, $1.5
Finished reading the file.
```

--- Virtual Vegetable Store ---

Available Vegetables:

1. Carrot (Orange) - \$0.75/lb
2. Spinach (Green) - \$1.25/lb
3. Brussel_sprouts (Green) - \$1.6/lb
4. Beet (Red) - \$0.9/lb
5. Eggplant (Purple) - \$1.5/lb
0. Exit

Choose a vegetable by number: 3

Enter pounds to purchase: 4

Profit on this transaction: \$4.4

--- Virtual Vegetable Store ---

Available Vegetables:

1. Carrot (Orange) - \$0.75/lb
2. Spinach (Green) - \$1.25/lb
3. Brussel_sprouts (Green) - \$1.6/lb
4. Beet (Red) - \$0.9/lb
5. Eggplant (Purple) - \$1.5/lb
0. Exit

Choose a vegetable by number: 4

Enter pounds to purchase: 4

Profit on this transaction: \$2.2

--- Virtual Vegetable Store ---

Available Vegetables:

1. Carrot (Orange) - \$0.75/lb
2. Spinach (Green) - \$1.25/lb
3. Brussel_sprouts (Green) - \$1.6/lb
4. Beet (Red) - \$0.9/lb
5. Eggplant (Purple) - \$1.5/lb
0. Exit

Choose a vegetable by number: 9

Invalid selection.

--- Virtual Vegetable Store ---

Available Vegetables:

1. Carrot (Orange) - \$0.75/lb
2. Spinach (Green) - \$1.25/lb
3. Brussel_sprouts (Green) - \$1.6/lb
4. Beet (Red) - \$0.9/lb
5. Eggplant (Purple) - \$1.5/lb
0. Exit

Choose a vegetable by number: 1
Enter pounds to purchase: 6
Profit on this transaction: \$2.7

--- Virtual Vegetable Store ---

Available Vegetables:

1. Carrot (Orange) - \$0.75/lb
2. Spinach (Green) - \$1.25/lb
3. Brussel_sprouts (Green) - \$1.6/lb
4. Beet (Red) - \$0.9/lb
5. Eggplant (Purple) - \$1.5/lb
0. Exit

Choose a vegetable by number: 0
Thank you for shopping!