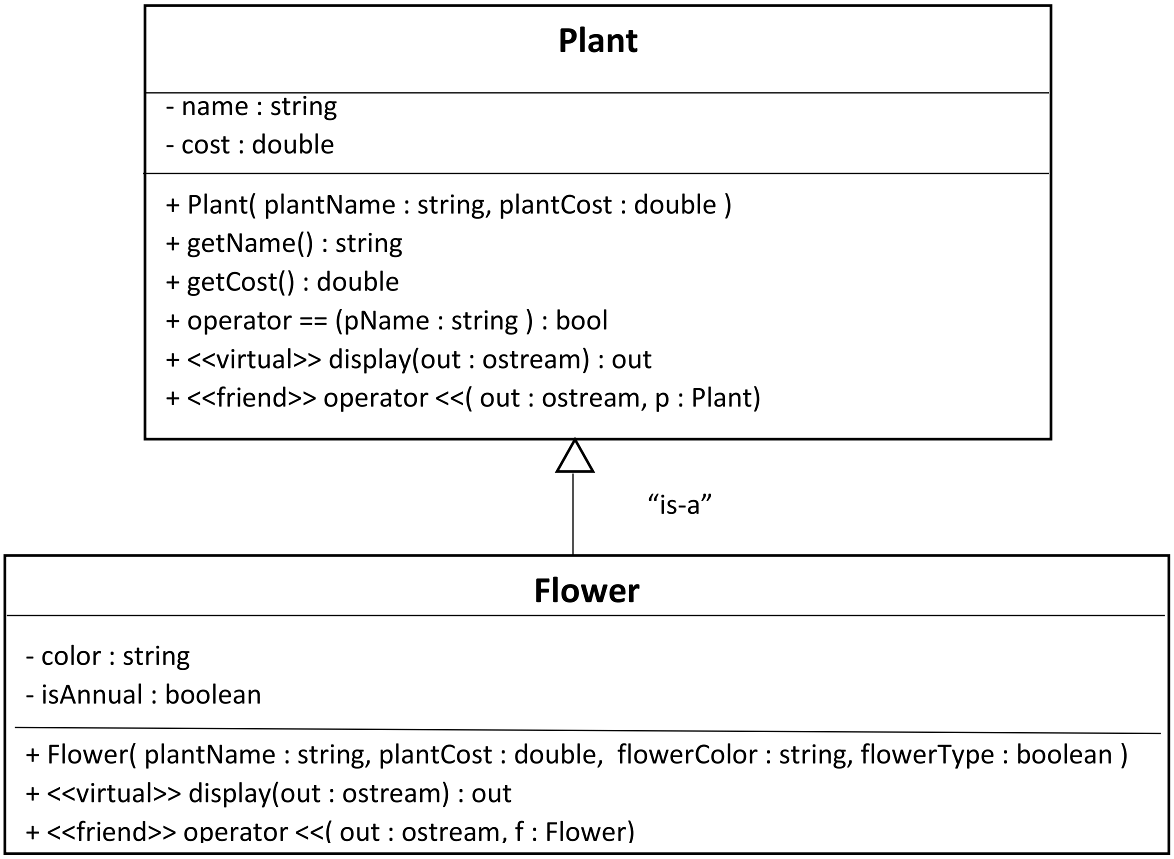
CSC240 – Lab 6

Inheritance, Polymorphism & STL

Points: 50 points

Implement the following hierarchy of classes:



Plant

- name : string
- cost : double

+ Plant(plantName : string, plantCost : double)
+ getName() : string
+ getCost() : double
+ operator == (pName : string) : bool
+ <<virtual>> display(out : ostream) : out
+ <<friend>> operator <<(out : ostream, p : Plant)

"is-a"

Flower

- color : string
- isAnnual : boolean

+ Flower(plantName : string, plantCost : double, flowerColor : string, flowerType : boolean)
+ <<virtual>> display(out : ostream) : out
+ <<friend>> operator <<(out : ostream, f : Flower)

Given a base class, Plant, and a derived class, Flower, write a menu driven program using a vector. The vector should be able to store pointers to objects that belong to the Plant class AND the Flower class. (This will enable *polymorphism*.)

Include functions to:

- addPlant – add a new Plant or Flower pointer to the vector. Prompt the user for information.
- removePlant – remove a Plant or Flower from the vector
    - write a loop using an iterator that starts with begin() and increments through the vector looking for a string that matches the name of the Plant the user inputs. Once the loop locates the element in the vector (using overloaded == operator), you can call the erase method. Otherwise display an error message.

```
vector<Dish*>::iterator ptr;
for(ptr = meal.begin(); ptr != meal.end(); ptr++)
    if( (**ptr) == "ham" )
        break;

    if( ptr == meal.end())
        cout << "ham not found\n";
    else
        cout << "ham was found\n";
```

(**ptr) explanation:

The first dereference is to derefence the iterator to get the Dish* from the vector.

The second dereference is to dereference the pointer to get the Dish object.

Order of operations require ( ) to be sure all the dereferencing is done before the ==.

use vector erase() method

```
iterator erase (iterator position);
iterator erase (iterator first, iterator last);
```

`erase()` removes from the vector either a single element (*position*) or a range of elements (`[first,last)`).

- display – display the contents of the vector
- total cost – display the total cost of all Plants and Flowers in the vector
- sort by plant name
  - Use the STL `sort()` algorithm to sort all plants and flowers by name.

    void sort (RandomAccessIterator first, RandomAccessIterator last,

    ```
                                                  Compare compFunction);
    ```

    compFunction is a helper function that takes two items as arguments and compares those items. The function returns true if the first element should appear <u>before</u> the second element when sorted, and false otherwise.

- sort by plants cost
  - Use the STL `sort()` algorithm to sort all plants and flowers by price. Write a helper function to compare prices.

- quit – program should terminate

Continue displaying the menu of options after each user selection until the user selects “quit”.

SAMPLE OUTPUT

```
***MY GARDEN MENU***
   1.  Add a new plant
   2.  Remove a plant
   3.  Display plants
   4.  Total cost
   5.  Sort by plant name
   6.  Sort by plant cost
   7.  Quit

Enter your selection: 1

Enter plant type ('P' - plant, 'F' - flower): P
Enter plant name: Mint
Enter plant cost: $4.00
```

```
***MY GARDEN MENU***
   1.  Add a new plant
   2.  Remove a plant
   3.  Display plants
   4.  Total cost
   5.  Sort by plant name
   6.  Sort by plant cost
   7.  Quit

Enter your selection: 1

Enter plant type ('P' - plant, 'F' - flower): F
Enter plant name: Hibiscus
Enter plant cost: $12.99
Enter flower color: red
Enter if flower is an annual (Y/N)? N
```

```
***MY GARDEN MENU***
   1.  Add a new plant
   2.  Remove a plant
   3.  Display plants
   4.  Total cost
   5.  Sort by plant name
   6.  Sort by plant cost
   7.  Quit

Enter your selection: 3

Plant 1: Mint       $ 4.00
Plant 2: Hibiscus   $12.99      red     perennial
```

display “annual” if boolean value, `isAnnual`, is true, otherwise display “perennial”

```
***MY GARDEN MENU***
   1.  Add a new plant
   2.  Remove a plant
   3.  Display plants
   4.  Total cost
   5.  Sort by plant name
   6.  Sort by plant cost
   7.  Quit

Enter your selection: 4

The total cost of all plants and flowers is $16.99.
```

display output in a table format, consider using printf()?

To output a C++ string variable using printf(), myStr, it must be converted into a Cstring: `myStr.c_str()`

```
***MY GARDEN MENU***
   1.  Add a new plant
   2.  Remove a plant
   3.  Display plants
   4.  Total cost
   5.  Sort by plant name
   6.  Sort by plant cost
   7.  Quit
Enter your selection: 2

Enter plant name to remove: Pansy
ERROR – Pansy is not in the garden.
```

error message if plant is not in the vector

```
***MY GARDEN MENU***
   1.  Add a new plant
   2.  Remove a plant
   3.  Display plants
   4.  Total cost
   5.  Sort by plant name
   6.  Sort by plant cost
   7.  Quit

Enter your selection: 5

Plant 1: Hibiscus   $12.99      red     perennial
Plant 2: Mint       $ 4.00
```

```
***MY GARDEN MENU***
   1.  Add a new plant
   2.  Remove a plant
   3.  Display plants
   4.  Total cost
   5.  Sort by plant name
   6.  Sort by plant cost
   7.  Quit

Enter your selection: 6

Plant 1: Mint       $ 4.00
Plant 2: Hibiscus   $12.99      red     perennial
```

```
***MY GARDEN MENU***
   1.  Add a new plant
   2.  Remove a plant
   3.  Display plants
   4.  Total cost
   5.  Sort by plant name
   6.  Sort by plant cost
   7.  Quit

Enter your selection: 9
ERROR – Invalid selection.
```

error message if menu option is invalid

```
***MY GARDEN MENU***
   1.  Add a new plant
   2.  Remove a plant
   3.  Display plants
   4.  Total cost
   5.  Sort by plant name
   6.  Sort by plant cost
   7.  Quit

Enter your selection: 7

Program complete.
```

NOTE: We will limit plant names and colors to a single word. That will allow you to use the extraction operators (>>) to read user input. Anything with a multiple words (ie. includes a space) would use getline(cin, var) to read the whitespace. *This is optional.*