```
Script started on Thu 19 Apr 2012 12:26:07 AM CDT
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ p
wd
/home/georgia/cplusplus
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at flowers.info
Name: Jakob Hansen
Class: CSC121 - Evening Section
Lab: Flower shop project

Levels attempted:

4 - For just getting the basic instructions down.

+2 - For the random category generation.

+.5...maybe? I reused my suffix lab... but didn’t put it in a library (in the
interest of time, again.)

+ a pat on the back for trying the string storage idea and then re-writing the
entire program with a different structure?

Program Description:

A program that will allow the clerk at a flower store with a silly name to enter
in the orders for a "blossom blanket". Customers can choose up to five bunches
of flowers from 3 different categories of flowers. The program will calculate
the total price, as well as keep track of the inventory of each category of
flower. If the stock of a certain flower goes to 0 of course, the clerk is not
allowed to put it on the order. The program also prints dollar amounts pretty
and append the appropriate suffixes to ordinal numbers, whenever they are
printed.

\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at flowers.cpp
#include <iostream>
#include <string>
#include <cstdlib>
#include <climits>
#include <iomanip>

using namespace std;

string suffix(short number);

char readCategory(short num_categories);

bool stockChecker(short & quant, short & stock);

int main(void)
{
        const double pretty_price = 15.0,
                         soso_price = 8.5,
                         ugly_price = 1.8;

        double total_price;

        char choice, yesno;

        short num_categories,
              pretty_stock,
              soso_stock,
              ugly_stock,
              p_quant = 0,
              s_quant = 0,
              u_quant = 0,
              i;

        srand(time(NULL));
        cout << "\n\tWelcome to the Alliterative Flower Calculation Program!\n"
                "\nFirst, you’ll need to enter the quantities of all the flowers.\n
";

        cout << "\nHow many flowers in the Pretty Category?  ";
        cin >> pretty_stock;

        while ( cin.fail() )
        {
            cin.clear();
            cout << "\nInvalid character: ’"
                 << static_cast<char>(cin.peek())
                 << "’!\n\nTry again: ";
            cin.ignore();
            cin >> pretty_stock;
        }

        cout << "\nHow many flowers in the So-So Category?  ";
        cin >> soso_stock;

        while ( cin.fail() )
        {
            cin.clear();
            cout << "\nInvalid character: ’"
                 << static_cast<char>(cin.peek())
                 << "’!\n\nTry again: ";
            cin.ignore();
            cin >> soso_stock;
        }

        cout << "\nHow many flowers in the Ugly Category?  ";
        cin >> ugly_stock;

        while ( cin.fail() )
        {
            cin.clear();
            cout << "\nInvalid character: ’"
                 << static_cast<char>(cin.peek())
                 << "’!\n\nTry again: ";
            cin.ignore();
            cin >> ugly_stock;
        }


        do
        {
            p_quant = 0;
            s_quant = 0;
            u_quant = 0;

            cout << "Thank you. Please enter your order information.\n"
                    "\nHow many categories in this basket?  ";

            cin >> num_categories;

            while (num_categories < 1 || num_categories > 5 ||
                   cin.fail())
            {
                cin.clear();

                cout << "\nPlease enter only numbers between 1 and 5!\n"
                                "How many categories?  ";

                        cin.ignore();
                    cin >> num_categories;
                }

                i = 1;
            while (i <= num_categories)
            {
                choice = readCategory(i);

                if (choice == ’P’)
                {
                    if (!stockChecker(p_quant, pretty_stock))
                    {
                        cout << "\nSorry, I’m all out of Pretty Flowers. \n";

                    }
                    else
                    {
                        i++;
                    }
                }
                else if (choice == ’S’)
                {
                    if (!stockChecker(s_quant, soso_stock))
                    {
                        cout << "\nSorry, I’m all out of So so Flowers. \n";
                    }
                    else
                    {
                        i++;
                    }
                }
                else
                {
                    if (!stockChecker(u_quant, ugly_stock))
                    {
                        cout << "\nSorry, I’m all out of Ugly Flowers. \n";
                    }
                    else
                    {
                        i++;
                    }
                }
            }

            total_price = p_quant * pretty_price +
                      s_quant * soso_price +
                      u_quant * ugly_price;

        cout << "\nThe total price will be " ;

        cout.setf(ios_base::fixed);
        cout.setf(ios_base::showpoint);
        cout.precision(2);

    cout << "$" << total_price << "\n";


        cout << "\n\nYou now have: \n"
             << pretty_stock << " pretty flowers,\n"
             << soso_stock << " so so flowers, \n"
             << ugly_stock << " ugly flowers. \n";

        cout << "\nWould you like to enter another order?  ";
        cin >> yesno;

        cin.ignore(INT_MAX, ’\n’);

        }
        while(toupper(yesno) == ’Y’);

    cout << "\nThanks for using the program. Good-Bye!\n";
```

```
    return 0;
}


// Returns the correct suffix for ordinal numbers

string suffix(short number)
{
    short ones_place;
    short tens_place;
    string suffix;

    ones_place = number % 10;

    tens_place = number/10 % 10;

    if (tens_place == 1)
    {
        suffix = "th";
    }
    else if (ones_place == 1)
    {
        suffix = "st";
    }
    else if (ones_place == 2)
    {
        suffix = "nd";
    }
    else if (ones_place == 3)
    {
        suffix = "rd";
    }
    else
    {
        suffix = "th";
    }

    return suffix;
}

/* Reads in a char for user’s category choice. Takes the number of category the
loop is on as an argument, to make sure the ordinal suffixes are correct. Also
contains the random category generation if the user enters R.*/

char readCategory(short num_category)
{
    char choice;
    short num;

        cout << "\n What category for the " << num_category
             << suffix(num_category) + " bunch?  ";

    cin >> choice;

    cin.ignore(INT_MAX, ’\n’);

        while (toupper(choice) != ’P’ &&
               toupper(choice) != ’S’ &&
               toupper(choice) != ’U’ &&
               toupper(choice) != ’R’)
        {
            cout << "\n Please enter Pretty, So-so, Ugly, or Random for the "
                 << num_category << suffix(num_category) + " bunch.  ";

            cin >> choice;

            cin.ignore(INT_MAX, ’\n’);

        }
        if (toupper(choice) == ’R’)
        {
            num = rand() % 3 + 1;

            switch(num)
            {
                case 1:
                    choice = ’p’;
                    cout << "\nThe Randomly Chosen " << num_category
                         << suffix(num_category) << " Category is Pretty.\n";
                    break;
                case 2:
                    choice = ’s’;
                    cout << "\nThe Randomly Chosen " << num_category
                         << suffix(num_category) << " Category is So-So.\n";
                         break;
                case 3:
                    choice = ’u’;
                    cout << "\nThe Randomly Chosen " << num_category
                         << suffix(num_category) << " Category is Ugly.\n";
                         break;
            }
        }

        choice = toupper(choice);

        return choice;
}

/*After the char is read in, depending on what category it happens to be,
you pass the related quantity and stock amounts. Increments and decrements
those quantities as long as there IS stock. If not, it returns
false without changing any numbers. */

bool stockChecker(short & quant, short & stock)
{
    bool enough;


        if (stock >= 1)
        {
            quant++;
            stock--;
            enough = true;
        }


        else
        {
                enough = false;
        }

        return enough;
}

\033]0;georgia@georgia-MT6017: ˜/cplusplus\007georgia@georgia-MT6017:˜/cplusplus$ C
PP flowers
flowers.cpp***
flowers.cpp: In function â\200\230char readCategory(short int)â\200\231:
flowers.cpp:243:25: warning: conversion to â\200\230short intâ\200\231 from
â\200\230intâ\200\231 may alter its value [-Wconversion]
flowers.cpp:265:25: warning: conversion to â\200\230charâ\200\231 from â\200\230int
â\200\231
may alter its value [-Wconversion]


\033]0;georgia@georgia-MT6017: ˜/cplusplus\007georgia@georgia-MT6017:˜/cplusplus$ .
/flowers.out
```

```
        Welcome to the Alliterative Flower Calculation Program!

First, you’ll need to enter the quantities of all the flowers.

How many flowers in the Pretty Category?  10

How many flowers in the So-So Category?  10

How many flowers in the Ugly Category?  10
Thank you. Please enter your order information.

How many categories in this basket?  7

Please enter only numbers between 1 and 5!
How many categories?  t

Please enter only numbers between 1 and 5!
How many categories?  5

 What category for the 1st bunch?  r

The Randomly Chosen 1st Category is Ugly.

 What category for the 2nd bunch?  t

 Please enter Pretty, So-so, Ugly, or Random for the 2nd bunch.  p

 What category for the 3rd bunch?  s

 What category for the 4th bunch?  u

 What category for the 5th bunch?  r

The Randomly Chosen 5th Category is So-So.

The total price will be $35.60


You now have:
9 pretty flowers,
8 so so flowers,
8 ugly flowers.

Would you like to enter another order?  y
Thank you. Please enter your order information.

How many categories in this basket?  5

 What category for the 1st bunch?  r

The Randomly Chosen 1st Category is Pretty.

 What category for the 2nd bunch?  r

The Randomly Chosen 2nd Category is Ugly.

 What category for the 3rd bunch?  r

The Randomly Chosen 3rd Category is So-So.

 What category for the 4th bunch?  r

The Randomly Chosen 4th Category is So-So.

 What category for the 5th bunch?  r

The Randomly Chosen 5th Category is Ugly.
The total price will be $35.60


You now have:
8 pretty flowers,
6 so so flowers,
6 ugly flowers.

Would you like to enter another order?  n

Thanks for using the program. Good-Bye!
\033]0;georgia@georgia-MT6017: ˜/cplusplus\007georgia@georgia-MT6017:˜/cplusplus$ e
xit
exit

Script done on Thu 19 Apr 2012 12:27:46 AM CDT
```