```
Script started on Wed 17 Oct 2012 10:25:52 PM CDT
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at dublist.h
#ifndef DUBLIST_H_INC
#define DUBLIST_H_INC

#include <iostream>
#include <cstddef>

using namespace std;

class doublelist
{
    double * p;  // array of doubles
    size_t current_size;            // number of array positions filled
    size_t capacity;     // number of POSSIBLE array positions

    bool reallocate(size_t more);

public:


    doublelist(void) : p(NULL), current_size(0), capacity(0) { }

    doublelist(size_t caller_size) : p(new double[caller_size]),
                  current_size(0), capacity(caller_size)
    {
        capacity = p == NULL ? 0 : caller_size;
    }

    doublelist(const doublelist & dl);

    ~doublelist(void) { delete [] p; p = NULL; }

    doublelist & operator=(const doublelist & dl);

    void insert_double(double num);

    double get_last(void) { return p == NULL ? 0.0 : p[current_size-1]; }

    void delete_last(void)
    {
        if (p != NULL)
        {
            current_size--;
        }
        return;
    }

    size_t get_size(void) { return p == NULL ? 0 : current_size; }

    void print(void) const;
};

#endif
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at dublist.cpp
#include "dublist.h"
#include <cstddef>
#include <iostream>

using namespace std;


doublelist::doublelist(const doublelist & dl)
          : p(new double[dl.capacity]),
            capacity(dl.capacity),
            current_size(dl.current_size)
{
    if (p != NULL)
    {
        for (size_t i=0; i != current_size; i++)
        {
            p[i] = dl.p[i];
        }
    }
    else
    {
        current_size = capacity = 0;
    }
}

doublelist & doublelist::operator=(const doublelist & dl)
{
    if ( &dl != this )
    {
        delete [] p;

        p = new double[dl.capacity];

        if (p != NULL)
        {
            capacity = dl.capacity;
            current_size = dl.current_size;
            for(size_t i=0; i != current_size; i++)
            {
                p[i] = dl.p[i];
            }
        }
        else
        {
            current_size = capacity = 0;
        }
    }
    return *this;
}

bool doublelist::reallocate(size_t more)
{
    double * pnew;
    bool okay = false;
    pnew = new double [capacity + more];
    if (pnew != NULL)
    {
        okay = true;
        for (size_t i=0; i != current_size; i++)
        {
            pnew[i] = p[i];
        }
        delete [] p;
        p = pnew;
        pnew = NULL;
        capacity += more;
    }
    return okay;
}


void doublelist::insert_double(double num)
{
    if (current_size == capacity)
    {
        cout << "\nThe list has reached it’s capacity! Time to reallocate...";
        if (reallocate(capacity))
        {
            p[current_size++] = num;
            cout << "\nReallocation Successful!!";
            cout << "\nNew Capacity is " << capacity;
        }
        else
        {
            cout << "\nERROR: MEMORY ALLOCATION FAILED";
        }
    }
    else
    {
        p[current_size++] = num;
    }
    return;
}


void doublelist::print(void) const
{
    long i;
    for (i = 0; i < current_size; i++)
    {
        cout << p[i] << ' ';
    }
    return;
}

\033]0;georgia@georgia-MT6017: ˜/cplusplus\007georgia@georgia-MT6017:˜/cplusplus$ c
at doubletest.\033[Kcpp
#include "dublist.h"
#include <iostream>
#include <cstddef>
#include <climits>

using namespace std;

int main(void)
{
    double test;
    char yesno;
    size_t size;
    doublelist dlist, dlist2;

    cout << "\nDynamic list of Doubles!\n";

    cout << "\nWhat size do you want your list to be? ";

    cin >> size;

    cout << "\nOk, a list of size: " << size;

    dlist = doublelist(size);

    cout << "\nEnter a bunch of doubles! ";

    cin >> test;

    while (!cin.fail())
    {
        dlist.insert_double(test);
        cin >> test;
    }
    cin.clear();
    cin.ignore(INT_MAX, '\n');


    cout << "\nThe Current number of filled spots in dlist is: " << dlist.get_size(
);

    cout << "\nThe Last number in dlist is: " << dlist.get_last();

    cout << "\nDeleting last number in dlist - the size is: ";
```

```

    dlist.delete_last();

    cout << dlist.get_size()
         << "\nThe last number in dlist is now: " << dlist.get_last();

    cout << "\nThe whole dlist is: ";

    dlist.print();


    cout << "\nEnter in another size: ";

    cin >> size;

    cout << "\nYou entered: " << size;

    dlist2 = doublelist(size);

    cout << "\nNow dlist2 has a size of: " << size;

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

    cout << "\nEnter a bunch of doubles! ";

    cin >> test;

    while (!cin.fail())
    {
        dlist2.insert_double(test);
        cin >> test;
    }
    cin.clear();
    cin.ignore(INT_MAX, '\n');


    cout << "\nCopying dlist into dlist2...";

    dlist2 = dlist;

    cout << "\ndlist2 is now: ";

    dlist2.print();


    cout << "\n\nBye!";
    return 0;
}
\033]0;georgia@georgia-MT6017: ˜/cplusplus\007georgia@georgia-MT6017:˜/cplusplus$ C
PP doubletest dublist
doubletest.cpp***
dublist.cpp...
doubletest.cpp: In function â\200\230int main()â\200\231:
doubletest.cpp:11:10: warning: unused variable â\200\230yesnoâ\200\231
[-Wunused-variable]
dublist.h: In copy constructor â\200\230doublelist::doublelist(const
doublelist&)â\200\231:
dublist.h:13:12: warning: â\200\230doublelist::capacityâ\200\231 will be
initialized after [-Wreorder]
dublist.h:12:12: warning:   â\200\230size_t doublelist::current_sizeâ\200\231
[-Wreorder]
dublist.cpp:8:1: warning:   when initialized here [-Wreorder]
dublist.cpp: In member function â\200\230void doublelist::print() constâ\200\231:
dublist.cpp:99:21: warning: comparison between signed and unsigned
integer expressions [-Wsign-compare]


\033]0;georgia@georgia-MT6017: ˜/cplusplus\007georgia@georgia-MT6017:˜/cplusplus$ .
\033[K\033[Kdo\033[K\033[K\033[K./doubletest.out
Dynamic list of Doubles!

What size do you want your list to be? 5

Ok, a list of size: 5
Enter a bunch of doubles! 1.2 2.333.4 4.5 5.6 6.7 7.8q

The list has reached it’s capacity! Time to reallocate...
Reallocation Successful!!
New Capacity is 10
The Current number of filled spots in dlist is: 7
The Last number in dlist is: 7.8
Deleting last number in dlist - the size is: 6
The last number in dlist is now: 6.7
The whole dlist is: 1.2 2.3 3.4 4.5 5.6 6.7
Enter in another size: 4

You entered: 4
Now dlist2 has a size of: 4
Enter a bunch of doubles! 5.6 6.7 7.8 8.9 9.1 10.11 q

The list has reached it’s capacity! Time to reallocate...
Reallocation Successful!!
New Capacity is 8
Copying dlist into dlist2...
dlist2 is now: 1.2 2.3 3.4 4.5 5.6 6.7

Bye!\033]0;georgia@georgia-MT6017: ˜/cplusplus\007georgia@georgia-MT6017:˜/cplusplu
s$ exit
exit

Script done on Wed 17 Oct 2012 10:28:00 PM CDT
```