```
#include <iostream>
using std::cout;
using std::endl;

class CDemo
{
    int iNo;
    float fNo;
    double dNo;
    char chChar;

    int *pPtr1;
    int *pPtr2;

public:
    CDemo()
    {
        iNo = 0;
        fNo = 0.0f;
        dNo = 0.0;
        chChar = '\0';

        pPtr1 = NULL;

        pPtr2 = new int[3];
        if(NULL == pPtr2)
        {
            cout << "Memory allocation FAILED";
            return;
        }

        /*
        for(int iCounter = 0; iCounter < 3; iCounter++)
            pPtr2[iCounter] = 0;
        */

        memset(pPtr2, 0, 3 * sizeof(int));
    }

    void Set(int iParam, float fParam, double dParam, char chParam)
    {
        iNo = iParam;
        fNo = fParam;
        dNo = dParam;
        chChar = chParam;

        pPtr1 = &iNo;

        for(int iCounter = 0; iCounter < 3; iCounter++)
            pPtr2[iCounter] = iParam + iCounter + 1;
    }

    void Get()
    {
        cout << "iNo is " << iNo << endl;
        cout << "fNo is " << fNo << endl;
        cout << "dNo is " << dNo << endl;
        cout << "chChar is " << chChar << endl;

        cout << "&iNo is " << &iNo << endl;
        cout << "pPtr1 is " << pPtr1 << endl;
        if(pPtr1 != NULL)
            cout << "*pPtr1 is " << *pPtr1 << endl;

        cout << "pPtr2 is " << pPtr2 << endl;
        cout << "pPtr2 values are:\t";
        for(int iCounter = 0; iCounter < 3; iCounter++)
```

```
            cout << "pPtr2[" << iCounter << "] = " << pPtr2[iCounter] << "\t";

        cout << endl;
    }

    CDemo(CDemo &refObj)
    {
        cout << "\nIn copy contructor\n";

        iNo = refObj.iNo;
        fNo = refObj.fNo;
        dNo = refObj.dNo;
        chChar = refObj.chChar;

        pPtr1 = &iNo;
        pPtr2 = new int[3];
        if(NULL == pPtr2)
        {
            cout << "Memory allocation FAILED\n";
            return;
        }

        /*
        for(int iCounter = 0; iCounter < 3; iCounter++)
            pPtr2[iCounter] = refObj.pPtr2[iCounter];
        */

        memcpy(pPtr2, refObj.pPtr2, 3 * sizeof(int));
    }

    CDemo& operator =(CDemo &refOperand2)
    {
        cout << "\nIn assignment operator\n";

        iNo = refOperand2.iNo;
        fNo = refOperand2.fNo;
        dNo = refOperand2.dNo;
        chChar = refOperand2.chChar;

        pPtr1 = &iNo;

        if(NULL == pPtr2)   // In our case, pPtr2 surely NOT NULL
        {
            pPtr2 = new int[3];
            if(NULL == pPtr2)
            {
                cout << "Memory allocation FAILED\n";
                return *this;
            }
        }
        /*
        for(int iCounter = 0; iCounter < 3; iCounter++)
            pPtr2[iCounter] = refObj.pPtr2[iCounter];
        */

        memcpy(pPtr2, refOperand2.pPtr2, 3 * sizeof(int));

        return *this;
    }

    ~CDemo()
    {
        iNo = 0;
        fNo = 0.0f;
        dNo = 0.0;
        chChar = '\0';

        pPtr1 = NULL;

        if(pPtr2 != NULL)
        {
            delete pPtr2;
            pPtr2 = NULL;
        }
    }

};

int main(void)
{
    CDemo obj1;

    cout << "OBJ1 values are:\n";
    obj1.Get();

    obj1.Set(10, 57.33f, 69.33, 'A');
    cout << "\nOBJ1 values are:\n";
    obj1.Get();

    CDemo obj2 = obj1;
    cout << "\nOBJ2 values are:\n";
    obj2.Get();

    CDemo obj3;
    obj3 = obj1;
    cout << "\nOBJ3 values are:\n";
    obj3.Get();

    return 0;
}
```