

```
#include <iostream>
#include <string>

#include "array.h"

std::ostream& operator<<(std::ostream& s, const Array& rhs)
{
    for(int i=0; i < rhs.size(); ++i)
        std::cout << "[" << i << "] = " << rhs.arr[i] << std::endl;
}

const Array& Array::operator=(const Array& rhs)
{
    if(this != &rhs) { // standard alias test
        // copy all elements of rhs into this
        sz = rhs.sz;
        if(arr) delete arr;
        arr = new int [rhs.size()];
        for(int i=0; i<rhs.size(); i++) arr[i] = rhs.arr[i];
    }
}

Array::Array(int isz)
{
    // this constructor creates the Array object given a size
    sz = isz;
    arr = new int [sz];

    memset(arr,0,sz*sizeof(int));
}

Array::Array(int *array, int isz)
{
    // this constructor creates the Array object given a size
    // and source of data (namely, array)
    sz = isz;
    arr = new int [sz];

    for(int i=0; i < sz; ++i)
        arr[i] = array[i];
}

Array::Array(const Array& rhs)
{
    // this constructor creates the Array object given a constant
    // reference to another Array object (result is a copy of
    // the other object)
    sz = rhs.sz;
    arr = new int [sz];

    for(int i=0; i < sz; ++i)
        arr[i] = rhs.arr[i];
}
```

```
#include <iostream>
#include "array.h"
int main()
{
    Array  myArr1, myArr2(3), myArr3 = myArr1;
    Array  *myArrp = new Array;

    std::cout << "printing myArr1\n";
    std::cout << myArr1;

    std::cout << "printing myArr2\n";
    std::cout << myArr2;

    std::cout << "printing myArr3\n";
    std::cout << myArr3;

    delete myArrp;
}
```

Array