

You must create the following program, which has three files.

bfmain.h:

```
#include <iostream>
using namespace std;
#include "BigFour.h"

GenArray A;
GenArray MyFunc(GenArray Arg);

int main()
{
    cout<<"Starting main\n";
    GenArray B;
    cout<<"First assignment\n";
    A = 25;
    cout<<"Second assignment\n";
    B = A;
    cout<<"Function call\n";
    B = MyFunc(B);
    cout<<"Exiting Main\n";

    int zzz;
    cin>>zzz;
    return 0;
}

GenArray MyFunc(GenArray Arg)
{
    cout<<"Entering MyFunc\n";
    GenArray RetVal(6);

    for (int i=0 ; i<6 ; i++)
    {
        RetVal.SetVal(i*i,i); // Just to have something here
    }
    cout<<"Exiting MyFunc\n";
    return RetVal;
}
```

BigFour.h:

```
#ifndef BIGFOUR_H
#define BIGFOUR_H

class GenArray
{
private:
    int *Data;
    int Size;
public:
    // The "Big Four" required functions
    GenArray();
    ~GenArray();
    GenArray(const GenArray &Old);
    GenArray &operator=(const GenArray &Old);

    // Not really necessary
    void SetVal(int NewVal,int Index);
    int GetVal(int Index);

    // convert constructor
    GenArray(int NewSize);
};

#endif
```

BigFour.cpp:

```
#include "BigFour.h"
#include <iostream>
using namespace std;

GenArray::GenArray()
{
    cout<<"Default Constructor\n";
    Size = 10;
    Data = new int[Size];
    for (long i=0 ; i<Size ; i++)
    {
        Data[i] = 0;
    }
}

GenArray::~GenArray()
{
    cout<<"Destructor\n";
    delete Data;
}

GenArray::GenArray(const GenArray &Old)
{
    cout<<"Copy Constructor\n";
    Size = Old.Size;
    Data = new int[Size];
    for (long i=0 ; i<Size ; i++)
    {
        Data[i] = Old.Data[i];
    }
}

GenArray &GenArray::operator=(const GenArray &Old)
{
    cout<<"Assignment Overload\n";
    delete Data;
    Size = Old.Size;
    Data = new int[Size];
    for (long i=0 ; i<Size ; i++)
    {
        Data[i] = Old.Data[i];
    }
    return *this;
}

void GenArray::SetVal(int NewVal,int Index)
{
    if (Index >= 0 && Index < Size)
    {
        Data[Index] = NewVal;
    }
}

int GenArray::GetVal(int Index)
{
    if (Index >= 0 && Index < Size)
    {
        return Data[Index];
    }
    return 0;
}

GenArray::GenArray(int NewSize)
{
    cout<<"int Constructor\n";
    Size = NewSize;
    if (Size <= 0)
```

```
{
    Size = 10;
}
Data = new int[Size];
for (long i=0 ; i<Size ; i++)
{
    Data[i] = 0;
}
}
```

Once you have entered this code, run it, and print the results. If necessary copy and paste the output into the Windows Notepad. On this output will be several messages from the constructors and destructors. Next to each line, identify the variable that is being constructed or destroyed. There are several temporary variables. When you identify them write something like “Temporary variable for return value” next to them. For declared variables, identify the name of the variable itself.

Upload you code to the upload site so I can see your work.

Turn in the printed output with your notations.