```
// A safe array example.
#include <iostream>
#include <cstdlib>
#include<string.h>
using namespace std;


class atype{
    int ncols;
    int *dynamicArray;
    public:

    atype(){

        ncols=0;;
        dynamicArray = NULL;



    }
    //constructor
    atype(int col){

        ncols=col;
        dynamicArray = new int[ncols];


    }

    //destructor
    ~atype(){

       delete [] dynamicArray;
       dynamicArray=NULL;


    }

    //user inserting elements in 2d array
    void fillArray()
      {

                for (int in=0;in<ncols;++in){
                                int value;
                                cout<<"enter value";
                                cin>>value;
                                dynamicArray[in] = value;
                }

      }


    //bound checking-safe array implementation
    int &operator [](int i){
```

```
        if(i<0 || i>  ncols-1 ) {
            cout << "Boundary Error\n";
            exit(1);

        }
    return dynamicArray[i];
    }

    atype(const atype& rhs)   //copy constructor
    {

         ncols = rhs.ncols;

         dynamicArray = new int[ncols];


         memcpy(dynamicArray,rhs.dynamicArray, sizeof(int)*ncols);

    }

    atype& operator=(const atype& rhs)    //assignment operator
      {
            if (this == &rhs)
                  return *this;

        delete[] dynamicArray;
        dynamicArray = NULL;


        ncols = rhs.ncols;
        dynamicArray = new int[ncols];
        memcpy(dynamicArray,rhs.dynamicArray, sizeof(int)*ncols);


         return *this;
      }







};




int main()
{
      int columns;

      cout<<"enter cols"<<endl;
      cin>>columns;
```

```
        atype ob1(columns);

        ob1.fillArray();

        atype ob2=ob1;

        cout << ob1[1] << endl;
        cout<<ob2[2]<<endl;     //checking bounds of array




        return 0;
}
```