

Another stack example

Another example of a stack using a pre-defined fixed-capacity array.

```
#include<iostream>
using namespace std;
const int MAX_CAPACITY=128;

class A{
public:
    int val[10];
    A(int s=1){
        for (int i=0; i<10;i++){
            val[i]=s+i*s;
        }
    }

    void print(){
        for (int i=0;i<10;i++){
            cout <<val[i]<<" ";
        }
        cout<<endl;
    }
};

class Stack{
public:
    int top; //top of the stack, bottom is 0
    A data[MAX_CAPACITY]; //stack is modeled as an array of A objects
    bool push(A); //push onto the stack - return success or failure
    A pop(); //pop from stack - return the top of the stack
    Stack(){}
```

You could potentially return a pointer to A, or nothing at all if you want to discard the popped object

```
};

bool Stack::push(A a){
    ++top;
    if (top<=MAX_CAPACITY){
        data[top]=a;
        return true;
    }else {
        return false;
    }
}

A Stack::pop(){
    A t=data[top];
    top--;
    return t;
}

int main(){
    A a1(4), a2(3),a3(8), a4(2), a5(6);
    Stack st;
    st.push(a1);
    st.push(a2);
    st.push(a5);
    st.push(a3);
    A b=st.pop();
    b.print();
    b=st.pop();
    b.print();
}

/*****Output*****/
igraine:~/PDS/C++$ ./stack
8 16 24 32 40 48 56 64 72 80
6 12 18 24 30 36 42 48 54 60
igraine:~/PDS/C++$
*****/
```

This shouldn't be top++. Why?