```
1  /*-------------------------------------------------------------------
2  Copyright (c) 2015 Author: Jagadeesh Vasudevamurthy
3  file: dstack.h
4  -------------------------------------------------------------------*/
5
6  /*-------------------------------------------------------------------
7  This file has dstack class declaration
8  -------------------------------------------------------------------*/
9
10 /*-------------------------------------------------------------------
11 All includes here
12 -------------------------------------------------------------------*/
13 #ifndef dstack_H
14 #define dstack_H
15
16 #include "../util/util.h"
17 #include "../darray/darray.h"
18
19 /*-------------------------------------------------------------------
20 Declaration of dstack class
21 -------------------------------------------------------------------*/
22 template <typename T>
23 class dstack {
24 public:
25   explicit dstack(int capacity = 50, bool display = false);
26   explicit dstack(bool);
27   explicit dstack(bool d, int c) = delete;
28   ~dstack();
29   int num_elements() const;
30   bool isempty() const;
31   bool isfull() const;
32   void push(const T& b); // Stack copies b and holds. Now stack is the owner of b
33   T& top(); // user can get top by alias. He can change its contents also. See explanation in
       implementation
34   void pop(); // Remove top element from the stack. Nothing returned
35
36   void for_each_element_of_stack_from_top_to_bottom(void(*pf) (T& c));
37   bool display()const { return _display; }
38   void set_display(bool x) {
39     darray<T>::set_display(x);
40     _display = x;
41   }
42   /* no body will copies or equal stack */
43   dstack(const dstack<T>& s) = delete;
44   dstack<T>& operator=(const dstack<T>& rhs) = delete;
45 private:
46   bool _display;
47   int _sp;
48   darray<T> _stack;
49 };
50
51 #include "dstack.hpp"
52
53 #endif
54 //EOF
55
56
```