```
2 // stackcpp.h
3 // A stack implementation in C++.
  // Author: Lauren E. Scott
5 // June 27, 2014
  //
7
  8
  #include <iostream>
  #include <stdlib.h>
10
  #include <stdbool.h>
11
12
  using namespace std;
13
14
  template <class T>
15
  class Stack {
  public:
17
     Stack(int s) { array = new T[s]; size = s; contains = 0; }
18
     ~Stack() {}
19
20
     void push(T data);
21
     T pop();
22
     T top();
23
     bool isEmpty();
24
     bool isFull();
25
     bool hasTwoElements();
26
     void print();
27
28
29
30
  private:
     T* array;
31
     int size, contains;
32
33
  };
34
  // Function: push
  // Push an item onto the top of the stack.
  38
39
  template <class T>
40
  void Stack<T>::push(T data) {
41
42
     if(isFull()) {
43
         cout << "Stack is full. " << endl;</pre>
44
         cout << "Size: " << size << endl;</pre>
45
```

```
cout << "Contains: " << contains << endl;</pre>
46
47
         return;
     }
48
     array[contains] = data;
49
     contains++;
50
     cout << "Successfully pushed " << data << endl;</pre>
51
52
  }
53
54
55
  // Function: pop
56
  // Pop an item off of the top of the stack, returning that item.
  58
59
  template <class T>
60
  T Stack<T>::pop() {
61
     if(isEmpty()) {
62
         cout << "Stack is empty. " << endl;</pre>
63
     } else {
64
         T result = array[contains-1];
65
         delete array[contains];
                                     // Set current position to null.
  //
66
         [array]contains = 0;
  //
67
         contains--:
68
         return result:
69
70
     }
  }
71
72
  73
  // Function: top
74
  // Access the top item of the stack without returning it.
  76
77
78
  template <class T>
  T Stack<T>::top() {
79
     if(isEmpty()) {
80
         cout << "Stack is empty. " << endl;</pre>
81
         return -1;
82
  //
     } else {
83
         return array[contains-1];
84
     }
85
  }
86
87
  88
  // Function: isEmpty
  // Checks to see whether the stack (thus, underlying array) is empty.
```

```
92
  template <class T>
93
  bool Stack<T>::isEmpty() {
     if (contains <= 0) {</pre>
95
        return true:
96
     }
97
     return false;
98
  }
99
100
  101
  // Function: isFull
102
  // Checks to see whether the stack (thus, underlying array) is full
  105
  template <class T>
106
  bool Stack<T>::isFull() {
107
     if (contains >= size) {
108
        return true:
109
     }
110
     return false;
111
112
  }
113
  114
  // Function: hasTwoElements
  // Checks to see whether the stack has at least two elements.
116
  // A specific function written for the stack game.
  118
119
  template <class T>
120
  bool Stack<T>::hasTwoElements() {
121
     return (contains >= 2);
122
123
  }
124
  125
126 // Function: print
  // Prints the contents of the stack.
  128
129
  template <class T>
130
  void Stack<T>::print() {
131
     for(int i = contains-1; i >= 0; i--) {
132
        cout << " | " << array[i];</pre>
133
     }
134
     cout << " ||" << endl;
135
```

/Users/laurenscott/Documents/Codi...cture_Games/Stack_Game/stackcpp.h Page 4 of 4 Saved: 6/30/14, 10:23:13 PM Printed For: Lauren Scott

136 } 137

138