

```
1  /* * * * * * * * * * * * * * * * * * * * * * * * * * * */
2  // stack_game.cpp
3  // A game using my stack implementation in C++.
4  // Author: Lauren E. Scott
5  // June 27, 2014
6  //
7  /* * * * * * * * * * * * * * * * * * * * * * * * * * * */
8
9  #include "stackcpp.h"
10
11 int calculate(Stack<string>* s, string input) {
12     int result = 0;
13     size_t sz;
14     if(input == "+" || input == "-" || input == "/" || input == "*") {
15         if(!(s->hasTwoElements())) {
16             cout << "Stack is empty." << endl;
17             return 1;
18         } else {
19             string top = s->pop();
20             cout << "Top :" << top << endl;
21             string second = s->pop();
22             cout << "Second :" << second << endl;
23             if(input == "+") {
24                 result = stoi(top) + stoi(second);
25                 cout << "Result: " << result << endl;
26                 s->push(to_string(result));
27             } else if(input == "-") {
28                 result = stoi(top) - stoi(second);
29                 cout << "Result: " << result << endl;
30                 s->push(to_string(result));
31             } else if (input == "/") {
32                 if (second == "0") {
33                     cout << "Can't divide by 0. " << endl;
34                     return 1;
35                 } else {
36                     result = stoi(top) / stoi(second);
37                     cout << "Result: " << result << endl;
38                     s->push(to_string(result));
39                 }
40             } else if (input == "*") {
41                 result = stoi(top) * stoi(second);
42                 cout << "Result: " << result << endl;
43                 s->push(to_string(result));
44             }
45         }
46     }
```

```
46     } else if(stoi(input) >= 48 || stoi(input) < 58) { // I
47         s->push(input); // .
48     } else {
49         cout << "Please input only an operator (+, -, *, /) or an integer!"
50         return 1;
51     }
52     return 0;
53 }
54
55 void serve_level(Stack<string>* s, string answer, string allowable_numbers)
56     string calculated_answer = "0";
57     string player_in = "";
58
59 //     cout << answer << endl;
60 //     cout << "Available numbers: " << allowable_numbers << endl;
61     cout << "Push integers and operands onto the list to create the solution
62     while(calculated_answer != answer) {
63         cin >> player_in;
64         calculate(s, player_in);
65         calculated_answer = s->top();
66         cout << "Current stack: Top --> ";
67         s->print();
68 //         cout << "Answer: " << answer << endl;
69 //         cout << "Top calculated answer: " << calculated_answer << endl;
70
71     }
72     cout << "Nice! Answer was " << answer << endl;
73 }
74
75 void initialize_levels(string levels[5]) {
76     levels[0] = "12";
77     levels[1] = "3";
78     levels[2] = "100";
79     levels[3] = "56";
80     levels[4] = "0";
81 }
82
83 void initialize_available_numbers(string available_numbers[5]) {
84     available_numbers[0] = "6214";
85     available_numbers[1] = "5643";
86     available_numbers[2] = "253";
87     available_numbers[3] = "24654";
88     available_numbers[4] = "0";
89 }
90
```

```
91 int main() {
92     string input, enter;
93     string levels[5];
94     string available_numbers[5];
95
96     initialize_levels(levels);
97     initialize_available_numbers(available_numbers);
98
99     cout << "----- The Stack Game -----" << endl;
100    cout << "Welcome to the stack game! Use a stack and Reverse Polish Notat
101    cout << "calculations to come up with the solution given in each level!"
102    cout << "[Press any key and ENTER]" << endl;
103    cin >> enter;
104
105    for (int i = 0; i < 5; i++) {
106        Stack<string> s(10);
107        cout << "----- Level " << i << "-----" << endl;
108        cout << "Produce this solution: " << levels[i] << endl;;
109        serve_level(&s, levels[i], available_numbers[i]);
110    }
111 }
112
113
114
115
116
117
118
119
120
121
122
```