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

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

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