

```

1 // this function is modeled after code found in regex_match_example.cpp
2 // note that this code demonstrates just one way to address the issue of parsing
3 // a command line using regular expressions, other approaches are not only
4 // possible, but perhaps even better
5 void test2_BasicCommandParsing_v1() {
6
7     string strCmd[10] ;
8     strCmd[0] = " add element root first one" ;
9     strCmd[1] = " add element root second" ;
10    strCmd[2] = " add attribute first attr1 attr1value" ;
11    strCmd[3] = " add attribute second attr2" ;
12    strCmd[4] = "print" ;
13    strCmd[5] = "a" ;
14    strCmd[6] = "ad" ;
15    strCmd[7] = "add" ;
16    strCmd[8] = "quit" ;
17    strCmd[9] = "another command" ;
18    int nCmds = 10 ;
19
20    cmatch what;
21    // what[0] contains the entire matched string
22    // what[1] contains the first matched group
23    // what[2] contains the second matched group
24    // what[3] etc.
25
26    regex reAddCmd( "\\s*a(d|dd)?.*", regex::icase ) ;
27    regex rePrintCmd( "\\s*p(r|ri|rin|rint)?.*", regex::icase ) ;
28    regex reQuitCmd( "\\s*q(u|ui|uit)?.*", regex::icase ) ;
29
30    // loop through all hard-coded command strings for testing purposes
31    for ( int n = 0 ; n < nCmds ; n++ ) {
32
33        // user entry point
34        cout << "\nYour command: " ;
35        // cin >> strCmd ;
36        cout << strCmd[n] << endl ;
37
38        // string version of a matched group
39        // for building a bridge between the cmatch type and an STL sting so that we can
40        // process matches with STL string functions
41        string strWhat ;
42
43        // test for a match of an ADD command
44        if ( regex_match( strCmd[n].c_str(), what, reAddCmd ) ) {
45            cout << " Command is ADD" << endl ;
46            cout << " Call a function to do your add command processing here." << endl ;
47        }
48
49        // test for a match of a PRINT command
50        else if ( regex_match( strCmd[n].c_str(), what, rePrintCmd ) ) {
51            cout << " Command is PRINT" << endl ;
52            cout << " Call your print function here." << endl ;
53        }
54
55        // handle a QUIT command
56        else if ( regex_match( strCmd[n].c_str(), what, reQuitCmd ) ) {
57            cout << " Command is QUIT" << endl ;
58            cout << " Goodbye." << endl ;
59            return ;
60        }
61
62        // parsing error: the first keyword is not ADD, PRINT, or QUIT
63        else {
64            cout << " Invalid command: 1st word must be 'add', 'print', or 'quit'." << endl ;
65        }
66    }
67 }

```