



- [Home](#)
  - [About](#)
  - [Business Plan »](#)
  - [Communication »](#)
  - [Dieting](#)
  - [Sales](#)
  - [Sitemap](#)
  - [Videos »](#)
  - [Web Design »](#)
- 
- [Communication »](#)
  - [Diet Nutritional](#)
  - [Flash Tutorial](#)
  - [How To »](#)
  - [Investing](#)
  - [iPad »](#)
  - [Marketing »](#)
  - [Most Popular](#)
  - [Royalty Free Photos](#)
  - [Sales](#)
  - [Web Design »](#)

[share](#)

## C++ Tutorial 13

Posted by [Derek Banas](#) on Apr 30, 2018 in [C Video Tutorial](#) | [0 comments](#)

In this tutorial we explore advanced uses of functions using C++. We'll cover how to store functions as variables, pass functions into other functions, store functions in a vector and more. Then we'll solve 2 problems. We'll receive lists of numbers and then return a list of only the odd values. We'll then generate random lists of heads and tails and add up how often each occurs. All of the code and a transcript of the video follows below.

If you like tutorials like this, consider [donating \\$1](#), or simply turn off Ad Blocking software. Either helps me to continue making these tutorials for free.



## Code & Transcript

```
1  #include <cstdlib>
2  #include <iostream>
3  #include <string>
4  #include <vector>
5  #include <ctime>
6  #include <numeric>
7  #include <cmath>
8  #include <sstream>
9
10 // 1. Created to store as a variable and to
11 // pass into other functions
12 double MultBy2(double num){
13     return num * 2;
14 }
15
16 // 2. Functions can receive other functions
17 // the first double is the return type followed
18 // by the data types for the parameter
19 double DoMath(std::function<double(double)> func, double num){
20     return func(num);
21 }
22
23 // 3. You can store functions in a vector
24 double MultBy3(double num){
25     return num * 3;
26 }
27
28 // 4. ----- PROBLEM -----
29 // Checks for odd using modulus
30 bool IsItOdd(int num){
31     if(num % 2 == 0){
32         return false;
33     } else {
34         return true;
```

```
35     }
36 }
37
38 // Receives a list and generates a list of odd values
39 // from that list
40 std::vector<int> ChangeList(std::vector<int> list,
41     std::function<bool(int)> func){
42     std::vector<int> oddList;
43     for(auto i: list){
44         if(func(i)){
45             oddList.push_back(i);
46         }
47     }
48     return oddList;
49 }
50 // ----- 4. END OF PROBLEM -----
51
52 // ----- 5. PROBLEM -----
53 // Generates a random list from the possible values supplied
54 std::vector<char> GetHAndTList(std::vector<char> possibleValues,
55     int numberValuesToGenerate){
56     srand(time(NULL));
57     std::vector<char> hAndTList;
58
59     for(int x = 0; x < numberValuesToGenerate; ++x){
60         int randIndex = rand() % 2;
61         hAndTList.push_back(possibleValues[randIndex]);
62     }
63     return hAndTList;
64 }
65
66 // Receives a list and sums the number of matching chars
67 int GetNumberOfMatches(std::vector<char> list,
68     char valueToFind){
69     int numOfMatches = 0;
70
71     for(char c: list){
72         if(c == valueToFind){
73             numOfMatches++;
74         }
75     }
76     return numOfMatches;
77 }
78
79 // ----- 5. END OF PROBLEM -----
80
81 int main()
82 {
83     // 1. You can store functions as variables
84     auto times2 = MultBy2;
85     std::cout << "5 * 2 = " <<
86         times2(5) << "\n";
87
88     // 2. Pass a function into a function
89     std::cout << "6 * 2 = " <<
90         DoMath(times2, 6) << "\n";
91
92     // 3. You can store functions in a vector
93     // Create a vector using the function parameters
94     // and then load the functions into the vector
95     std::vector<std::function<double(double)>> funcs (2);
96     funcs[0] = MultBy2;
97     funcs[1] = MultBy3;
98     std::cout << "2 * 10 = " <<
99         funcs[0](10) << "\n";
```

```
100
101 // 4. ----- PROBLEM -----
102 // Create a function that receives a list and a function
103 // The function passed will return True or False if a list
104 // value is odd.
105 // The surrounding function will return a list of odd
106 // numbers
107 std::vector<int> listOfNums {1,2,3,4,5};
108 std::vector<int> oddList = ChangeList(listOfNums,
109     IsItOdd);
110 std::cout << "List of Odds\n";
111 for(auto i: oddList){
112     std::cout << i << "\n";
113 }
114 // ----- 4. END OF PROBLEM -----
115
116 // ----- 5. PROBLEM -----
117 // Create a function that creates a random list using
118 // a limited number of values
119 // Create another function that checks for the number
120 // of matches in a list
121 // Create a random list of Hs and Ts and then output
122 // how many of each were generated
123
124 std::vector<char> possibleValues {'H', 'T'};
125 std::vector<char> hAndTList = GetHAndTList(possibleValues,
126     100);
127 std::cout << "Number of Heads : " <<
128     GetNumberOfMatches(hAndTList, 'H')
129     << "\n";
130 std::cout << "Number of Tails : " <<
131     GetNumberOfMatches(hAndTList, 'T')
132     << "\n";
133
134 // ----- 5. END OF PROBLEM -----
135
136
137 return 0;
138 }
```

## Leave a Reply

Your email address will not be published.

Comment




Name

Email

Website

Search

Social Networks

 Facebook YouTube Twitter LinkedIn

Buy me a Cup of Coffee

"Donations help me to keep the site running. One dollar is greatly appreciated." - (Pay Pal Secured)

 Donate

My Facebook Page

6K people like this. Be the first  
of your friends.

Archives

- [October 2018](#)
- [September 2018](#)
- [August 2018](#)
- [July 2018](#)
- [June 2018](#)
- [May 2018](#)
- [April 2018](#)
- [March 2018](#)
- [February 2018](#)
- [January 2018](#)
- [December 2017](#)
- [November 2017](#)
- [October 2017](#)
- [September 2017](#)
- [August 2017](#)
- [July 2017](#)
- [June 2017](#)
- [May 2017](#)
- [April 2017](#)
- [March 2017](#)
- [February 2017](#)
- [January 2017](#)
- [December 2016](#)
- [November 2016](#)
- [October 2016](#)
- [September 2016](#)
- [August 2016](#)
- [July 2016](#)
- [June 2016](#)
- [May 2016](#)
- [April 2016](#)
- [March 2016](#)
- [February 2016](#)

- [January 2016](#)
- [December 2015](#)
- [November 2015](#)
- [October 2015](#)
- [September 2015](#)
- [August 2015](#)
- [July 2015](#)
- [June 2015](#)
- [May 2015](#)
- [April 2015](#)
- [March 2015](#)
- [February 2015](#)
- [January 2015](#)
- [December 2014](#)
- [November 2014](#)
- [October 2014](#)
- [September 2014](#)
- [August 2014](#)
- [July 2014](#)
- [June 2014](#)
- [May 2014](#)
- [April 2014](#)
- [March 2014](#)
- [February 2014](#)
- [January 2014](#)
- [December 2013](#)
- [November 2013](#)
- [October 2013](#)
- [September 2013](#)
- [August 2013](#)
- [July 2013](#)
- [June 2013](#)
- [May 2013](#)
- [April 2013](#)
- [March 2013](#)
- [February 2013](#)
- [January 2013](#)
- [December 2012](#)
- [November 2012](#)
- [October 2012](#)
- [September 2012](#)
- [August 2012](#)
- [July 2012](#)
- [June 2012](#)
- [May 2012](#)
- [April 2012](#)
- [March 2012](#)
- [February 2012](#)
- [January 2012](#)
- [December 2011](#)
- [November 2011](#)
- [October 2011](#)
- [September 2011](#)
- [August 2011](#)

- [July 2011](#)
- [June 2011](#)
- [May 2011](#)
- [April 2011](#)
- [March 2011](#)
- [February 2011](#)
- [January 2011](#)
- [December 2010](#)
- [November 2010](#)
- [October 2010](#)
- [September 2010](#)
- [August 2010](#)
- [July 2010](#)
- [June 2010](#)
- [May 2010](#)
- [April 2010](#)
- [March 2010](#)
- [February 2010](#)
- [January 2010](#)
- [December 2009](#)

Powered by [WordPress](#) | Designed by [Elegant Themes](#)  
[About the Author](#) [Google+](#)