

thispointer.com

[C++11 Tutorials](#)[Python](#)[C++ Tutorials](#)[STL](#)[Multithreading](#)[Data Science](#)[Boost Library](#)[Design Patterns](#)[java](#)[GDB](#)[Datastructure](#)[Subscribe](#)[Home](#) » [C++ 11](#) » [delete](#) » You are reading »

C++11 / C++14 : 'delete' keyword and deleted functions with Use Cases | Examples

 Varun  August 25, 2018  C++ 11, delete  1 Comment

In this article we will discuss how to use delete keyword with functions and its use cases.

Deleting Functions using 'delete' keyword

In C++11 a new feature of keyword **delete** is introduced. Now we can apply the delete keyword to functions to make them uncallable.

```
1 void someFunction() = delete ;
```

Its practical use cases are,

- Deleting compiler generated functions like copy constructor, assignment operators, move constructor , move assignment operator and default constructor.
- Deleting member functions to prevent data loss conversions

[Std::Tuple](#)[Lambda Functions](#)[C++11 Utilities](#)[Initializer_list](#)[Std::Array](#)[Rvalue References](#)[Smart Pointers](#)[Multithreading](#)[Unordered_Set](#)[Unordered_Map](#)

[c++11 : std::tuple Tutorial](#)
[c++11 : std::make_tuple Tutorial](#)

Advertisements

Recent Articles

[Python : Convert list of lists or nested list to flat list](#)
September 23, 2019

- Restrict Object creation on Heap by deleting new operator for class
- Delete specific template specilaizations

Let's discuss them one by one i.e.

Deleting Copy Constructor and Assignment Operator

```

1  class User
2  {
3
4      int id;
5      std::string name;
6  public:
7      User(int userId, std::string userName) : id(userId), name
8      {}
9
10     // Copy Constructor is deleted
11     User(const User & obj) = delete;
12     // Assignment operator is deleted
13     User & operator = (const User & obj) = delete;
14
15     void display()
16     {
17         std::cout<<id << " ::: " <<name<<std::endl;
18     }
19
20 };

```

In the above class User, Copy Constructor and Assignment Operator are deleted. If anybody tries to call the deleted function then it will generate compile time error.

For example,

```

1  User userObj(3, "John");
2
3  User obj = userObj;

```

As copy constructor is deleted, therefore above lines will throw following compile time error i.e.

```

1  delete_example.cpp: In function 'int main()':
2  delete_example.cpp:136:13: error: use of deleted function 'Us
3      User obj = userObj;
4          ^
5  delete_example.cpp:111:2: note: declared here
6      User(const User & obj) = delete;

```

[Python: How to find all indexes of an item in a List?](#)
September 16, 2019

[Pandas : Select first or last N rows in a Dataframe using head\(\) & tail\(\)](#) September 14, 2019

[Python : Count elements in a list that satisfy certain conditions](#) September 8, 2019

[Python : Get number of elements in a list, lists of lists or nested list](#) September 7, 2019

[Pandas : Check if a value exists in a DataFrame using in & not in operator | isin\(\)](#) September 1, 2019

[Pandas : Change data type of single or multiple columns of Dataframe in Python](#) August 31, 2019

[Pandas : Get unique values in columns of a Dataframe in Python](#) August 19, 2019

[Python : min\(\) function Tutorial with examples](#) August 8, 2019

[Python : List Comprehension vs Generator expression explained with examples](#) August 6, 2019

[Pandas : Drop rows from a dataframe with missing values or NaN in columns](#) August 4, 2019

[Python : max\(\) function explained with examples](#) July 20, 2019

[Python : Iterators vs Generators](#) July 17, 2019

[How to Merge two or more Dictionaries in Python ?](#) July 14, 2019

[Docker : How to Stop &](#)

A part from applying delete keywords to compiler generated functions, we can apply delete keyword to other functions too.

Deleting member functions to prevent data loss conversions

Some times due to automatic type conversion we can call the functions with wrong parameters too. For example, for User class we have this parameterized constructor i.e.

```
1 User(int userId, std::string userName) : id(userId), name(use
2 {}
```

Even if this constructor accepts id as int, still we can call this by double 7 char too i.e.

```
1 User obj4(5.5, "Riti");
2
3 User obj5('a', "Riti");
```

It will work but these are invalid conversions, with delete keyword we can prevent them too by declaring them deleted i.e.

```
1 // Deleting a constructor that accepts a double as ID to preve
2 User(double userId, std::string userName) = delete ;
3
4 // Deleting a constructor that accepts a double as ID to preve
5 User(char userId, std::string userName) = delete ;
```

Now on creating User object with double and char will throw compile time error i.e.

```
1 error: use of deleted function 'User::User(double, std::__cx
2     User obj4(5.5, "Riti");
3         ^
4 note: declared here
5     User(double userId, std::string userName) = delete ;
6
7 error: use of deleted function 'User::User(char, std::__cxx1
8     User obj5('a', "Riti");
9         ^
10 note: declared here
11     User(char userId, std::string userName) = delete ;
```

Restrict Object creation on Heap by

[Remove a running container by ID or Name ?](#) July 13, 2019

[Python : How to get all keys with maximum value in a Dictionary](#) July 9, 2019

[Python : How to make a class Iterable & create Iterator Class for it ?](#) July 6, 2019

[How to sort a Numpy Array in Python ?](#) July 4, 2019

[Python : Yield Keyword & Generators explained with examples](#) June 29, 2019

[What is a Structured Numpy Array and how to create and sort it in Python?](#) June 28, 2019

[Python : Iterator, Iterable and Iteration explained with examples](#) June 26, 2019

[How to Stop & remove all or running or stopped docker containers ?](#) June 24, 2019

[Count occurrences of a single or multiple characters in string and find their index positions](#) June 22, 2019

[How to create multi line string objects in python ?](#) June 20, 2019

[How to save Numpy Array to a CSV File using numpy.savetxt\(\) in Python](#) June 15, 2019

Advertisements

Popular Categories

[Behavioral Design Patterns \(6\)](#)

[Boost Date Time Library \(6\)](#)

[Boost Library \(17\)](#) **C++**

deleting new operator for class

We can delete the new operator of class User i.e.

```
1 class User
2 {
3     int id;
4     std::string name;
5 public:
6     User(int userId, std::string userName) : id(userId), name(userName) {}
7
8     // Delete the new function to prevent object creation on heap
9     void * operator new (size_t) = delete;
10
11 };
12
```

Now if we try to create User Object on heap using new, it will give compile time error i.e.

```
1 User * ptr = new User(1, "Riti");
```

Compile Error:

```
1 error: use of deleted function ‘static void* User::operator new (size_t)’
2     User * ptr = new User(1, "Riti");
3                          ^
4 note: declared here
5     void * operator new (size_t) = delete;
```

Complete example is as follows,

```
1 #include <iostream>
2 #include <string>
3
4 class User
5 {
6
7     int id;
8     std::string name;
9 public:
10     User(int userId, std::string userName) : id(userId), name(userName) {}
11
12     // Copy Constructor is deleted
13     User(const User & obj) = delete;
14     // Assignment operator is deleted
15     User & operator = (const User & obj) = delete;
16
17 }
```

(107)C++ 11 (59)

c++11 Threads (16) C++ Interview Questions

(30) collections (17) Data

Science (10) Debugging (6)

Debugging Tutorial (6) Design

Patterns (9) dictionary (15)

Directories (13)

FileHandling (14) Functions

(10) gdb (6) gdb commands (6)

gdb Tutorial (6) HashSet (8)

Iterators (5) java (29) linkedlist

(5) Linux (8) Linux System

Programming (8) List (20)

Method Overriding (5)

Multithreading (12)

Numpy (22) Pandas

(35) Python

(146) Smart Pointers (6)

std::list (12) std::map (16)

std::set (10) std::string (12)

std::thread (5) std::vector (18)

STL (56) STL Algorithm (8)

STL Interview Questions

(16) strings (11)

Uncategorized (11)

unordered_map (8)

unordered_set (6)

Articles with most comments

About Us (17)

C++11 Multithreading - Part 7:

Condition Variables

Explained (9)

STL Tutorials and Interview

Questions (8)

C++11 Lambda Function :

```

18 void display()
19 {
20     std::cout<<id << " :: " <<name<<std::endl;
21 }
22
23 // Deleting a constructor that accepts a double as ID to prevent copy
24 User(double userId, std::string userName) = delete ;
25
26 // Deleting a constructor that accepts a double as ID to prevent copy
27 User(char userId, std::string userName) = delete ;
28
29 // Delete the new function to prevent object creation on heap
30 void * operator new (size_t) = delete;
31
32 };
33
34
35 int main()
36 {
37     User userObj(3, "John");
38
39     // Can not copy User object as copy constructor is deleted
40     //User obj = userObj;
41
42     /*
43     * Creating User objects with double or char as ID will cause compilation error
44     */
45     User obj4(5.5, "Riti");
46
47     obj4.display();
48
49     User obj5('a', "Riti");
50
51     obj5.display();
52     */
53
54     // Can not create object on heap as new operator is deleted
55     //User * ptr = new User(1, "Riti");
56
57
58     return 0;
59 }
60

```

Delete specific template specialisation

With delete keyword we can restrict certain template specialisations of template classes or functions. Let's see how to do that, Suppose we have a template class of Complex Number

```

1 template <typename T>
2 class ComplexNumber
3 {
4     T x;
5     T y;
6 public:
7     ComplexNumber(T a, T b) : x(a) , y(b)
8     {}
9     void display()

```

[Example & Tutorial \(7\)](#)

[C++ : Map Tutorial Part 1:](#)

[Usage Detail with](#)

[examples \(6\)](#)

[Designing Callbacks in C++ -](#)

[Part 1: Function Pointers \(5\)](#)

[C++11 Multithreading – Part](#)

[9: std::async Tutorial &](#)

[Example \(5\)](#)

[What is Hashing and Hash](#)

[Table? \(4\)](#)

[C++11 Move Constructor &](#)

[rvalue References \(4\)](#)

[C++11 : How to Stop or](#)

[Terminate a Thread \(4\)](#)

[Binary Tree Traversal with](#)

[Strategy design pattern and](#)

[open-closed principle \(3\)](#)

[Set vs Map : How to choose a](#)

[right associative container](#)

[? \(3\)](#)

[Difference between lvalue](#)

[and rvalue in C++ \(3\)](#)

[C++11 Multithreading - Part 8:](#)

[std::future , std::promise and](#)

[Returning values from](#)

[Thread \(3\)](#)

[C++11 Lambda : How to](#)

[capture local variables inside](#)

[Lambda ? \(3\)](#)

[C++11 Smart Pointer - Part 1:](#)

[shared_ptr Tutorial and](#)

[Examples \(3\)](#)

[std::bind - Tutorial and](#)

[Examples \(3\)](#)

[Python : How to add / append](#)

[key value pairs in dictionary](#)

[using dict.update\(\) \(3\)](#)

[Reverse a Singly Linked List in](#)

[a Single Iteration \(2\)](#)


```

10     {
11         std::cout<<x << " + i"<<y<<std::endl;
12     }
13 };

```

We can use ComplexNumber with different types like int, double & char etc. i.e.

```

1 ComplexNumber<int> obj1(1,2);
2
3 ComplexNumber<double> obj2(1.0,2.0);
4
5 ComplexNumber<char> obj3('1' , '2');

```

Now we want to restrict that no one can use ComplexNumber class with double or char as template parameter. For that we need to delete certain specialized functions i.e.

```

1 ComplexNumber(char a, char b) = delete;
2
3 ComplexNumber(double a, double b) = delete;

```

Now on trying to use ComplexNumber<> class with double or char as template parameter will generate compile time error.

Complete ComplexNumber class with deleted specialisations is as follows,

```

1 template <typename T>
2 class ComplexNumber
3 {
4     T x;
5     T y;
6 public:
7     ComplexNumber(T a, T b) : x(a) , y(b)
8     {}
9     void display()
10    {
11        std::cout<<x << " + i"<<y<<std::endl;
12    }
13    // Deleted template specialisation
14    ComplexNumber(char a, char b) = delete;
15    // Deleted template specialisation
16    ComplexNumber(double a, double b) = delete;
17 };

```

Different between deleted function and private functions

[C++11 Multithreading – Part 5: Using mutex to fix Race Conditions \(2\)](#)

[Learn State Design Pattern by Designing a Simple Music System \(2\)](#)

[C++11 Multithreading - Part 1 : Three Different ways to Create Threads \(2\)](#)

[Multithreading \(2\)](#)

[C++11 Multithreading – Part 10: packaged_task<> Example and Tutorial \(2\)](#)

[C++11 : std::tuple Tutorial & Examples \(2\)](#)

Advertisements

Related Articles

[C++11 : make_tuple Tutorial & Example July 26, 2017](#)

[Python Tuple : Append , Insert , Modify & delete elements in Tuple August 18, 2018](#)

[Python : How to find an element in Tuple by value August 11, 2018](#)

[Python Tuple: Different ways to create a tuple and Iterate over it August 5, 2018](#)

[Python: How to sort a list of tuples by 2nd Item using Lambda Function or Comparator February 25, 2018](#)

[Find the index of value in Numpy Array using numpy.where\(\) December 15,](#)

By making functions private we can also restrict its calling but still marking function deleted with delete keyword has its own advantages like,

- Private member functions can be called from other member functions, whereas, deleted functions can not be called even from other member functions.
- Deleted functions exists in name lookup , if compiler finds a function is deleted then it will not lookup for other matching functions based on type lookups, hence prevents unnecessary data loss and bugs.

[Join LinkedIn Group of Python Professional Developers](#) who wish to expand their network and share ideas.

You can also follow us On Twitter :

[Click Here to Subscribe for more Articles / Tutorials like this.](#)

Related Posts:

- [C++11 : How to use std::thread as a member variable in class ?](#)
- [Need of Operator Overloading in C++ ? | Tutorial and Examples](#)
- [Python : How to remove a file if exists and handle errors | os.remove\(\) | os.unlink\(\)](#)
- [C++11 Smart Pointer - Part 6 : unique_ptr Tutorial and Examples](#)
- [Python : How to delete a directory recursively using shutil.rmtree\(\)](#)
- [Different ways to Remove a key from Dictionary in Python |](#)

2018

[numpy.where\(\) : Tutorial & Examples | Python](#) January 1, 2019

[Python : *args | How to pass multiple arguments to function ?](#) July 22, 2018

[Find max value & its index in Numpy Array | numpy.amax\(\)](#) December 22, 2018

[Python Numpy : Create a Numpy Array from list, tuple or list of lists using numpy.array\(\)](#) November 25, 2018

[C++11 / C++14 : 'delete' keyword and deleted functions with Use Cases | Examples](#) August 25, 2018

[numpy.amin\(\) | Find minimum value in Numpy Array and it's index](#) January 27, 2019

[Pandas : 6 Different ways to iterate over rows in a Dataframe & Update while iterating row by row](#) March 9, 2019

[Python : Check if a String contains a sub string & find it's index | case insensitive](#) April 7, 2019

[C++11 : std::array Tutorial and examples](#) November 19, 2017

[Python : How to unpack list, tuple or dictionary to Function arguments using * & **](#) July 21, 2018

[Python : How to Sort a Dictionary by key or Value ?](#) June 9, 2018

[Python : How to get all keys](#)

[del vs dict.pop\(\)](#)

- [pandas.apply\(\): Apply a function to each row/column in Dataframe](#)
- [Python : How to Remove multiple keys from Dictionary while Iterating ?](#)
- [C++ Map : Operator \[\] - Usage Details](#)
- [C++11 : std::tuple Tutorial & Examples](#)
- [Delete elements, rows or columns from a Numpy Array by index positions using numpy.delete\(\) in Python](#)
- [Pandas : How to create an empty DataFrame and append rows & columns to it in python](#)
- [Delete elements from a Numpy Array by value or conditions in Python](#)
- [C++ : How to copy / clone a STL List or Sub List](#)
- [Python : How to remove element from a list by value or Index | remove\(\) vs pop\(\) vs del](#)
- [C++11 : make_tuple Tutorial & Example](#)
- [Python : How to remove files by matching pattern | wildcards | certain extensions only ?](#)
- [map vs unordered_map | When to choose one over another ?](#)
- [C++: How to extract file extension from a path string using Boost & C++17 FileSystem Library](#)
- [C++11 - Variadic Template Function | Tutorial & Examples](#)
- [Unary Operator Overloading in C++ using both member & Friend function](#)
- [Python : How to pad strings with zero, space or some other character ?](#)
- [C++ : How to Sort a List of objects with custom Comparator or lambda function](#)
- [C++ : Convert double to string and manage precision | scientific notation](#)

with maximum value in a Dictionary July 9, 2019

Python : How to convert a list to dictionary ? June 30, 2018

Python Pandas : How to convert lists to a dataframe September 25, 2018

C++ : Convert double to string and manage precision | scientific notation June 29, 2017

Python : Iterator, Iterable and Iteration explained with examples June 26, 2019

Python : Find unique values in a numpy array with frequency & indices | numpy.unique() December 30, 2018

Python : 6 Different ways to create Dictionaries March 11, 2018

Pandas : count rows in a dataframe | all or those only that satisfy a condition March 2, 2019

Advertisements

Articles with recent comments

Biswajit Das on Python

Pandas : Drop columns in DataFrame by label Names or by Index Positions

jackzhao on C++11 Smart Pointer – Part 2: shared_ptr and Custom Deletor

Ionut Mihai on About Us

Nigel Blanchard on Python :

- [C++: How to get filename from a path with or without extension | Boost | C++17 FileSystem Library](#)

1 Comment Already



keshaw - October 3rd, 2018 at 1:16 pm

Excellent Definition.

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

This
site

Name *

Email *

Website

Post Comment

uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

« [Python Tuple : Append , Insert , Modify & delete elements in Tuple](#)

[Python Pandas : How to drop rows in DataFrame by index labels](#) »

[How to copy files from one location to another using shutil.copy\(\)](#)

[john on How to Merge two or more Dictionaries in Python ?](#)

[Leonard Inkret on map vs unordered_map | When to choose one over another ?](#)

[Python : List Comprehension vs Generator expression explained with examples –](#)

[thispointer.com on Python : Yield Keyword & Generators explained with examples](#)

[Kratika Srivastava on C++11 Smart Pointer – Part 6 : unique_ptr<> Tutorial and Examples](#)

[Aurelio on Select Rows & Columns by Name or Index in DataFrame using loc & iloc | Python Pandas](#)

[Prashant on About Us](#)

[Leshan Thomas on Pandas: Apply a function to single or selected columns or rows in Dataframe](#)

[Geri Reshef on pandas.apply\(\): Apply a function to each row/column in Dataframe](#)

[Wasin on What is std::deque and how deque works internally.](#)

[Michael Mixon on Python: How to create a zip archive from multiple files or Directory](#)

[Varun on How to change current working directory in python ?](#)

[Andre on How to change current working directory in](#)

[python ?](#)

Ashish on [Python : How to add / append key value pairs in dictionary using dict.update\(\)](#)

brelny on [C++ : How to compare two vectors | std::equal\(\) & Comparators](#)

Sean Flanigan on [About Us](#)

Kris Lakshmana on [Python Pandas : How to create DataFrame from dictionary ?](#)

Michael on [C++11 Move Contsructor & rvalue References](#)

Nick on [Python Pandas : How to add rows in a DataFrame using dataframe.append\(\) & loc\[\] , iloc\[\]](#)

tobias on [Pandas : count rows in a dataframe | all or those only that satisfy a condition](#)

Jyn K on [Python Pandas : How to create DataFrame from dictionary ?](#)

Akshat Jain on [std::bind – Tutorial and Examples](#)

Advertisements

Search

Subscribe with us

Subscribe For

* indicates required

Email Address *

Subscribe

Python : List

- Check if an item exists in List
- Check if a list contains all the elements of other list
- Create a List and initialize with values
- How to Iterate over a List
- Insert an element at specific index in List
- Sort a list of tuples by 2nd Item
- Sort a list of strings
- Add an element in list | append() vs extend()
- Check if all elements in a List are same
- Merge / Join two or more lists
- How to find all indexes of an item in a List ?
- Count elements in a list, lists of lists or nested list
- Remove Duplicates from a List
- Convert a list to string
- Remove element from a list by value or Index
- Remove multiple elements from list
- Convert list of lists or nested list to flat list
- Count elements in a

Pandas

- Create DataFrame from dictionary in pandas
- Get list of column and row names in DataFrame
- Change Column & Row names in DataFrame
- Select Rows & Columns using loc & iloc in DataFrame
- Select Rows based on conditions
- Drop rows in DataFrame by index labels
- Drop rows in DataFrame by conditions on column values
- Drop columns in DataFrame by label Names or Position
- Add new columns in a dataframe
- How to add rows in a DataFrame
- Count NaN or missing values in DataFrame
- Convert lists to a dataframe
- Find & Drop duplicate columns in a DataFrame
- Create an empty DataFrame and add

C++11 – Utilities

- Deleting Functions using ‘delete’ keyword
- std::bind
- auto specifier
- Variadic Templates

C++11 Smart Pointers

- shared_ptr<> Tutorial and Examples
- shared_ptr and Custom Deletor
- shared_ptr vs raw pointer
- Create shared_ptr objects carefully
- weak_ptr Tutorial | shared_ptr and Cyclic References
- unique_ptr<> Tutorial and Examples

C++11 – Unordered Set

- 1.) unordered_set Basic Example
- 2.) Initializing an unordered_set
- 3.) Inserting elements in an unordered_set
- 4.) Searching an

STL – Vector

- 1.) What’s std::vector and why to use it?
- 2.) Different ways to initialize a vector
- 3.) How does std::vector works internally
- 4.) User Defined Objects & std::vector
- 5.) How to use vector efficiently in C++?
- 6.) std::vector and Iterator Invalidation
- 7.) Remove repeated elements from a vector
- 8.) Fill a vector with random numbers
- 9.) Hidden cost of std::vector
- 10.) Adding elements in Vector

STL – Deque

- 1.) What is std::deque and how deque works internally.
- 2.) deque vs vector : What to choose ?

STL – List

- 1.) std::list Internals &

list that satisfy certain conditions

Python : Dictionary

Creating Dictionaries in Python

Iterating over Dictionaries in Python

Check if a key exists in Dictionary

Get list of all the keys in Dictionary

Get list of all the Values in a Dictionary

Remove multiple keys in Dictionary while Iterating

Remove a key from Dictionary

Add key/value pairs in Dictionary

Find keys by value in Dictionary

Sort a Dictionary by key or Value

Copy a dictionary | Shallow vs Deep Copy

Convert a list to dictionary

Filter a dictionary by conditions on keys or values

Python : How to get all keys with maximum value in a Dictionary

How to Merge two or more Dictionaries in Python ?

Python Strings

Access characters in string by index in Python

Iterate over the characters in string

How to Replace characters in a string ?

data to it later

Pandas : Check if a DataFrame is empty in Python

Pandas : Find duplicate rows in a Dataframe based on all or selected columns using DataFrame.duplicated() in Python

Pandas : skip rows while reading csv file to a Dataframe using read_csv() in Python

pandas.apply(): Apply a function to each row/column in Dataframe

Pandas: Sort rows or columns in Dataframe based on values using DataFrame.sort_values()

Pandas: Find maximum values & position in columns or rows of a Dataframe

Pandas : How to Merge Dataframes using DataFrame.merge() in Python – Part 1

Pandas : Merge Dataframes on specific columns or on index in Python – Part 2

Pandas : How to merge Dataframes by index using DataFrame.merge() – Part 3

Pandas : count rows in a dataframe | all or those only that satisfy a condition

Pandas : Read csv file to Dataframe with custom delimiter in Python

Pandas : 6 Different ways to iterate over rows in a Dataframe & Update while iterating row by row

Pandas : Loop or Iterate over all or certain columns of a dataframe

element in unordered_set

5.) unordered_set – Custom Hasher & Comparator

6.) Unordered_set & User defined classes

C++11 – UnorderedMap

Basic Usage Detail and Example

Initializing an unordered_map

Searching in unordered_map

Insert elements in unordered_map

Erasing an element

Erase elements while iterating

std::map vs std::unordered_map

C++11 Multithreading

Part 1: Three Ways to Create Threads

Part 2: Joining and Detaching Threads

Part 3: Passing Arguments to Threads

Part 4 : Sharing Data & Race Conditions

Part 5 : Fixing Race Conditions using mutex

Part 6 : Need of Event Handling

Part 7: Condition Variables

Part 8: std::future and std::promise

Part 9: std::async Tutorial & Example

Part 10: std::packaged_task<> Tutorial

Usage Details

2.) List vs Vector

3.) Different ways to Initialize a list

4.) Erase elements using iterators

5.) Remove elements while Iterating

6.) Remove elements based on External Criterion

7.) Get element by index in List

8.) Searching an element in std::list

9.) Different Ways to iterate over a List

10.) Sorting a List & custom Comparator

STL – Set

1.) C++ Set basic example and Tutorial

2.) Using std::set with user defined classes

3.) std::set and external Sorting criteria | Comparator

4.) Access Element by index in Set

5.) How to insert elements in Set

6.) How to iterate over a Set

7.) Removing an element from Set

8.) Erase elements while Iterating & Generic erase_if()

STL – Map

1.) std::map Usage Detail with examples

2.) std::map and Comparator

3.) std::map & User defined class objects

How to remove characters from a string by Index ?

How to pad strings with zero, space or some other character in Python

How to Compare Strings ? | Ignore case | regex | is vs == operator using Python

Find occurrence count & all indices of a sub-string in another string | including overlapping sub-strings

Check if a String contains a sub string & find it's index | case insensitive

Find frequency of each character in string and their indices | Finding duplicate characters in a string

How to create multi line string objects in python ?

Count occurrences of a single or multiple characters in string and find their index positions

Python : Functions

*args | Pass multiple arguments to function

**kwargs | Pass variable length key value pair as arguments to Function

Unpack list, tuple or dictionary to Function arguments

Why do we need Lambda functions in Python ? | Explained with examples.

How to use if, else if & else in Lambda Functions

How to use global variables in a function

Get minimum values in rows or columns & their index position in Dataframe using Python

Apply a function to single or selected columns or rows in Dataframe

Sort a DataFrame based on column names or row index labels using Dataframe.sort_index() in Pandas

Pandas : Drop rows from a dataframe with missing values or NaN in columns

Get unique values in columns of a Dataframe in Python

Change data type of single or multiple columns of Dataframe in Python

Check if a value exists in a DataFrame using in & not in operator | isin()

Select first or last N rows in a Dataframe using head() & tail()

Numpy

Create a numpy array from list, tuple or list of lists

Find the index of value in Numpy Array

Select element or sub array by index from numpy array

Select rows / columns by index from a 2D numpy array

Select elements by conditions from Numpy Array

Create a Numpy Array of evenly spaced numbers

C++11 Rvalue References

lvalue vs rvalue

Is rvalue immutable in C++?

What is rvalue reference in C++11

Move Constructor

Callbacks in C++

Function Pointers

Function Objects & Functors

Design Patterns

Behavioral Design Patterns

- ⋮ Observer Design Pattern
- ⋮ State Design Pattern
- ⋮ Strategy Design Pattern

Structural Design Patterns

- ⋮ Composite Design Pattern
- ⋮ Flyweight Design Pattern

Creational Design Patterns

Pointers

Pointer vs Reference

Allocating 2D Array Dynamically

C++ Strings

Find and Replace all occurrences of a string

Find all occurrences of a sub string

as keys

4.) Set vs Map

5.) How to Iterate over a map in C++

6.) Map Insert Example

7.) Iterate a map in reverse order

8.) Check if a key exists in a Map

9.) Search by value in a Map

10.) Erase by Key | Iterators

11.) C++ Map : Operator []

12.) Erase by Value or callback

13.) copy all Values from a Map to vector

STL Multimap

MultiMap Example and Tutorial

multimap::equals_range – Tutorial

STL Algorithms

std::sort Tutorial & Example

std::unique Tutorial & Example

1.) Using std::find & std::find_if with User Defined Classes

2.) Iterating over a range of User Defined objects and calling member function using std::for_each

C++ Memory Management

Memory Leaks

new and delete operator

Terms and Conditions Policy

Questions

Method Overriding
Tutorial

Overriding with
Different Return Type

Calling Base class's
overridden method

Preventing Method
Overriding

Need of preventing
Method Overriding

How to sort a Numpy
Array in Python ?

Python: Date Time Tutorials

Convert a timestamp
string to a datetime
object

Get Current date and
timestamp

Get Last Modification
date & time of a file