

good

Generated by Doxygen 1.13.2



<b>1 Class Index</b>	<b>1</b>
1.1 Class List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 LinkedList Class Reference	5
3.1.1 Constructor & Destructor Documentation	5
3.1.1.1 LinkedList()	5
3.1.1.2 ~LinkedList()	5
3.1.2 Member Function Documentation	6
3.1.2.1 clear()	6
3.1.2.2 copyFrom()	6
3.1.2.3 fillRandom()	6
3.1.2.4 print()	6
3.1.2.5 push()	6
3.1.2.6 sortMerge()	7
3.1.2.7 sortQuick()	7
3.2 Node Struct Reference	7
3.2.1 Constructor & Destructor Documentation	7
3.2.1.1 Node()	7
3.2.2 Member Data Documentation	8
3.2.2.1 data	8
3.2.2.2 next	8
<b>4 File Documentation</b>	<b>9</b>
4.1 LinkedList.cpp File Reference	9
4.2 LinkedList.h File Reference	9
4.3 LinkedList.h	9
4.4 main.cpp File Reference	10
4.4.1 Function Documentation	10
4.4.1.1 main()	10
<b>Index</b>	<b>11</b>



# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">LinkedList</a>	Singly Linked List class with sorting functionalities . . . . .	5
<a href="#">Node</a>	<a href="#">Node</a> structure for singly linked list . . . . .	7



# Chapter 2

## File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">LinkedList.cpp</a>	9
<a href="#">LinkedList.h</a>	9
<a href="#">main.cpp</a>	10





## Chapter 3

# Class Documentation

### 3.1 LinkedList Class Reference

```
#include <LinkedList.h>
```

#### Public Member Functions

- [LinkedList](#) ()
- [~LinkedList](#) ()
- void [push](#) (int data)
- void [print](#) () const
- void [clear](#) ()
- void [fillRandom](#) (int count)
- void [sortMerge](#) ()
- void [sortQuick](#) ()
- void [copyFrom](#) (const [LinkedList](#) &other)

#### 3.1.1 Constructor & Destructor Documentation

##### 3.1.1.1 [LinkedList\(\)](#)

```
LinkedList::LinkedList ()
```

Constructor.

##### 3.1.1.2 [~LinkedList\(\)](#)

```
LinkedList::~~LinkedList ()
```

Destructor.

## 3.1.2 Member Function Documentation

### 3.1.2.1 clear()

```
void LinkedList::clear ()
```

Clear the entire list.

### 3.1.2.2 copyFrom()

```
void LinkedList::copyFrom (  
    const LinkedList & other)
```

Deep copy the list from another instance.

#### Parameters

<i>other</i>	List to copy from
--------------	-------------------

### 3.1.2.3 fillRandom()

```
void LinkedList::fillRandom (  
    int count)
```

Fills the list with random integers.

#### Parameters

<i>count</i>	Number of elements to generate
--------------	--------------------------------

### 3.1.2.4 print()

```
void LinkedList::print () const
```

Print list.

### 3.1.2.5 push()

```
void LinkedList::push (  
    int data)
```

Insert new element at head.

#### Parameters

<i>data</i>	Value to insert
-------------	-----------------

### 3.1.2.6 sortMerge()

```
void LinkedList::sortMerge ()
```

Sorts the list using Merge Sort.

### 3.1.2.7 sortQuick()

```
void LinkedList::sortQuick ()
```

Sorts the list using Quick Sort.

The documentation for this class was generated from the following files:

- [LinkedList.h](#)
- [LinkedList.cpp](#)

## 3.2 Node Struct Reference

```
#include <LinkedList.h>
```

### Public Member Functions

- [Node](#) (int val)

### Public Attributes

- int [data](#)
- [Node](#) \* [next](#)

## 3.2.1 Constructor & Destructor Documentation

### 3.2.1.1 Node()

```
Node::Node (  
    int val)
```

Constructor.

#### Parameters

<i>val</i>	Initial value for node
------------	------------------------

## 3.2.2 Member Data Documentation

### 3.2.2.1 data

```
int Node::data
```

Data stored in the node.

### 3.2.2.2 next

```
Node* Node::next
```

Pointer to the next node.

The documentation for this struct was generated from the following files:

- [LinkedList.h](#)
- [LinkedList.cpp](#)

# Chapter 4

## File Documentation

### 4.1 LinkedList.cpp File Reference

```
#include "LinkedList.h"  
#include <cstdlib>
```

### 4.2 LinkedList.h File Reference

```
#include <iostream>
```

#### Classes

- struct [Node](#)
- class [LinkedList](#)

### 4.3 LinkedList.h

[Go to the documentation of this file.](#)

```
00001 #ifndef LINKEDLIST_H  
00002 #define LINKEDLIST_H  
00003  
00004 #include <iostream>  
00005  
00009 struct Node {  
00010     int data;  
00011     Node* next;  
00012  
00017     Node(int val);  
00018 };  
00019  
00023 class LinkedList {  
00024 private:  
00025     Node* head;  
00026  
00027     // Merge sort helpers  
00028     Node* mergeSort(Node* head);  
00029     Node* sortedMerge(Node* a, Node* b);  
00030     void frontBackSplit(Node* source, Node** frontRef, Node** backRef);
```

```
00031
00032 // Quick sort helpers
00033 Node* quickSort(Node* head);
00034 Node* quickSortRecur(Node* head, Node** newHead, Node** newEnd);
00035 Node* partition(Node* head, Node* end, Node** newHead, Node** newEnd);
00036
00037 public:
00041     LinkedList();
00042
00046     ~LinkedList();
00047
00052     void push(int data);
00053
00057     void print() const;
00058
00062     void clear();
00063
00068     void fillRandom(int count);
00069
00073     void sortMerge();
00074
00078     void sortQuick();
00079
00084     void copyFrom(const LinkedList& other);
00085 };
00086
00087 #endif // LINKEDLIST_H
```

## 4.4 main.cpp File Reference

```
#include <iostream>
#include <chrono>
#include "LinkedList.h"
```

### Functions

- int `main` ()

#### 4.4.1 Function Documentation

##### 4.4.1.1 `main()`

```
int main ()
```

Main function to test merge sort and quick sort with timing.

# Index

- [~LinkedList](#)
    - [LinkedList, 5](#)
- [clear](#)
  - [LinkedList, 6](#)
- [copyFrom](#)
  - [LinkedList, 6](#)
- [data](#)
  - [Node, 8](#)
- [fillRandom](#)
  - [LinkedList, 6](#)
- [LinkedList, 5](#)
  - [~LinkedList, 5](#)
  - [clear, 6](#)
  - [copyFrom, 6](#)
  - [fillRandom, 6](#)
  - [LinkedList, 5](#)
  - [print, 6](#)
  - [push, 6](#)
  - [sortMerge, 6](#)
  - [sortQuick, 7](#)
- [LinkedList.cpp, 9](#)
- [LinkedList.h, 9](#)
- [main](#)
  - [main.cpp, 10](#)
- [main.cpp, 10](#)
  - [main, 10](#)
- [next](#)
  - [Node, 8](#)
- [Node, 7](#)
  - [data, 8](#)
  - [next, 8](#)
  - [Node, 7](#)
- [print](#)
  - [LinkedList, 6](#)
- [push](#)
  - [LinkedList, 6](#)
- [sortMerge](#)
  - [LinkedList, 6](#)
- [sortQuick](#)
  - [LinkedList, 7](#)