

## PA04 Timer

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	binarySearch Class Reference . . . . .	7
4.1.1	Member Function Documentation . . . . .	7
4.1.1.1	operator()(int searchValue, const vector< int > &keys) const . . . . .	7
4.2	linearSearch Class Reference . . . . .	7
4.2.1	Member Function Documentation . . . . .	8
4.2.1.1	operator()(int searchValue, const vector< int > &keys) const . . . . .	8
4.3	Search Class Reference . . . . .	8
4.4	STLSearch Class Reference . . . . .	8
4.4.1	Member Function Documentation . . . . .	9
4.4.1.1	operator()(int searchValue, const vector< int > &keys) const . . . . .	9
4.5	TestVector Class Reference . . . . .	9
4.5.1	Constructor & Destructor Documentation . . . . .	9
4.5.1.1	TestVector(int size) . . . . .	9
4.5.1.2	TestVector(const TestVector &rhs) . . . . .	9
4.5.2	Member Function Documentation . . . . .	9
4.5.2.1	operator++() . . . . .	9
4.5.2.2	operator++(int ignored) . . . . .	9
4.5.2.3	operator[](int loc) const . . . . .	9
4.6	Timer Class Reference . . . . .	9
4.6.1	Constructor & Destructor Documentation . . . . .	10
4.6.1.1	Timer() . . . . .	10
4.6.2	Member Function Documentation . . . . .	10
4.6.2.1	getElapsedTime() const . . . . .	10
4.6.2.2	start() . . . . .	10
4.6.2.3	stop() . . . . .	11

<b>5 File Documentation</b>	<b>13</b>
5.1 config.h File Reference	13
5.1.1 Macro Definition Documentation	13
5.1.1.1 LAB13_TEST1	13
5.1.1.2 LAB13_TEST2	13
5.2 constructor.cpp File Reference	13
5.2.1 Macro Definition Documentation	14
5.2.1.1 runTest	14
5.2.2 Function Documentation	14
5.2.2.1 main(int argc, char **argv)	14
5.2.2.2 testCompute(DataType value)	14
5.2.2.3 testCompute< double >(double value)	14
5.2.2.4 testCompute< int >(int value)	14
5.2.2.5 testConstructor(int numValues, string name)	14
5.2.3 Variable Documentation	14
5.2.3.1 numRepetitions	14
5.3 inc.cpp File Reference	14
5.3.1 Function Documentation	15
5.3.1.1 main(int argc, char **argv)	15
5.3.2 Variable Documentation	15
5.3.2.1 numRepetitions	15
5.4 search.cpp File Reference	15
5.4.1 Function Documentation	15
5.4.1.1 main(int argc, char **argv)	15
5.4.2 Variable Documentation	15
5.4.2.1 numSearches	15
5.5 sort.cpp File Reference	15
5.5.1 Function Documentation	16
5.5.1.1 main(int argc, char **argv)	16
5.5.1.2 quickSort(vector< int >::iterator front, vector< int >::iterator back)	16

5.5.1.3	<a href="#">selectionSort(vector&lt; int &gt;::iterator front, vector&lt; int &gt;::iterator back)</a>	16
5.5.1.4	<a href="#">timeSort(void(*fcn)(vector&lt; int &gt;::iterator front, vector&lt; int &gt;::iterator back), const string name, const vector&lt; int &gt; &amp;masterList, const Timer &amp;overhead)</a>	16
5.5.2	<a href="#">Variable Documentation</a>	16
5.5.2.1	<a href="#">numSorts</a>	16
5.6	<a href="#">test13.cpp File Reference</a>	16
5.6.1	<a href="#">Function Documentation</a>	17
5.6.1.1	<a href="#">main()</a>	17
5.6.1.2	<a href="#">print_help()</a>	17
5.6.1.3	<a href="#">wait(int secs)</a>	17
5.7	<a href="#">testtimer.cpp File Reference</a>	17
5.7.1	<a href="#">Function Documentation</a>	17
5.7.1.1	<a href="#">getElapsed(timeval &amp;t1)</a>	17
5.7.1.2	<a href="#">main(int argc, char **argv)</a>	17
5.8	<a href="#">testvector.cpp File Reference</a>	17
5.9	<a href="#">testvector.h File Reference</a>	17
5.10	<a href="#">Timer.cpp File Reference</a>	18
5.11	<a href="#">Timer.h File Reference</a>	18
5.11.1	<a href="#">Detailed Description</a>	18
<b>Index</b>		<b>19</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

binary_function	
Search . . . . .	8
binarySearch . . . . .	7
linearSearch . . . . .	7
STLSearch . . . . .	8
TestVector . . . . .	9
Timer . . . . .	9





## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">binarySearch</a>	7
<a href="#">linearSearch</a>	7
<a href="#">Search</a>	8
<a href="#">STLSearch</a>	8
<a href="#">TestVector</a>	9
<a href="#">Timer</a>	9



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">config.h</a>	13
<a href="#">constructor.cpp</a>	13
<a href="#">inc.cpp</a>	14
<a href="#">search.cpp</a>	15
<a href="#">sort.cpp</a>	15
<a href="#">test13.cpp</a>	16
<a href="#">testtimer.cpp</a>	17
<a href="#">testvector.cpp</a>	17
<a href="#">testvector.h</a>	17
<a href="#">Timer.cpp</a>	18
<a href="#">Timer.h</a>	18

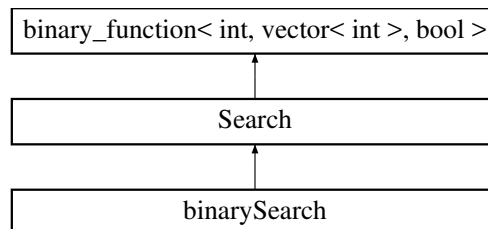


## Chapter 4

# Class Documentation

### 4.1 binarySearch Class Reference

Inheritance diagram for binarySearch:



#### Public Member Functions

- `bool operator() (int searchValue, const vector< int > &keys) const`

#### 4.1.1 Member Function Documentation

4.1.1.1 `bool binarySearch::operator() ( int searchValue, const vector< int > & keys ) const` `[inline], [virtual]`

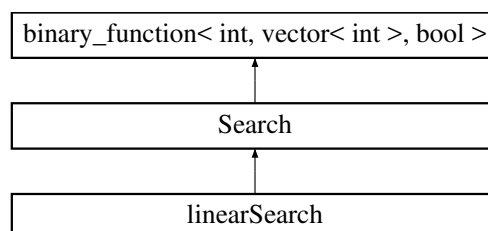
Implements [Search](#).

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

- [search.cpp](#)

### 4.2 linearSearch Class Reference

Inheritance diagram for linearSearch:



## Public Member Functions

- `bool operator() (int searchValue, const vector< int > &keys) const`

### 4.2.1 Member Function Documentation

4.2.1.1 `bool linearSearch::operator() ( int searchValue, const vector< int > & keys ) const` `[inline], [virtual]`

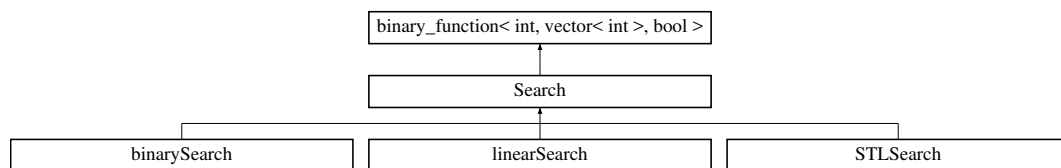
Implements [Search](#).

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

- [search.cpp](#)

## 4.3 Search Class Reference

Inheritance diagram for Search:

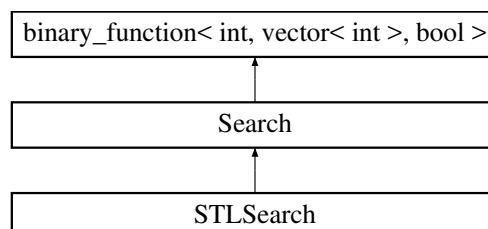


The documentation for this class was generated from the following file:

- [search.cpp](#)

## 4.4 STLSearch Class Reference

Inheritance diagram for STLSearch:



## Public Member Functions

- `bool operator() (int searchValue, const vector< int > &keys) const`

#### 4.4.1 Member Function Documentation

4.4.1.1 `bool STLSearch::operator() ( int searchValue, const vector< int > & keys ) const` `[inline], [virtual]`

Implements [Search](#).

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

- [search.cpp](#)

### 4.5 TestVector Class Reference

```
#include <testvector.h>
```

#### Public Member Functions

- [TestVector](#) (int *size*)
- [TestVector](#) (const [TestVector](#) &*rhs*)
- [TestVector](#) & [operator++](#) ()
- [TestVector](#) [operator++](#) (int *ignored*)
- int [operator\[\]](#) (int *loc*) const

#### 4.5.1 Constructor & Destructor Documentation

4.5.1.1 `TestVector::TestVector ( int size )`

4.5.1.2 `TestVector::TestVector ( const TestVector & rhs )`

#### 4.5.2 Member Function Documentation

4.5.2.1 `TestVector & TestVector::operator++ ( )`

4.5.2.2 `TestVector TestVector::operator++ ( int ignored )`

4.5.2.3 `int TestVector::operator[] ( int loc ) const`

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

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

### 4.6 Timer Class Reference

```
#include <Timer.h>
```

## Public Member Functions

- [Timer](#) ()
- void [start](#) () throw (runtime\_error)
- void [stop](#) () throw (logic\_error)
- double [getElapsedTime](#) () const throw (logic\_error)

### 4.6.1 Constructor & Destructor Documentation

#### 4.6.1.1 Timer::Timer ( )

<Constructor>

##### Parameters

None	
------	--

##### Precondition

None

##### Postcondition

timerWasStarted is set to 0

### 4.6.2 Member Function Documentation

#### 4.6.2.1 double Timer::getElapsedTime ( ) const throw logic\_error)

##### Returns

None  
[Timer](#) has been started and stopped  
Length of time that has passed has been calculated

#### 4.6.2.2 void Timer::start ( ) throw runtime\_error)

<Marks the beginning of a time interval (starts the timer)>

##### Parameters

None	
------	--

##### Precondition

None



**Postcondition**

[Timer](#) has been started

**Exceptions**

<i>Requires</i>	the clock function is working correctly.
-----------------	--

**4.6.2.3 void Timer::stop ( ) throw logic\_error)**

<Marks the end of a timer interval (stops the timer)>

**Parameters**

<i>None</i>	
-------------	--

**Precondition**

[Timer](#) has started

**Postcondition**

[Timer](#) has been stopped @ exception Requires the beginning of a time interval has been marked.

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

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



## Chapter 5

# File Documentation

### 5.1 config.h File Reference

#### Macros

- `#define LAB13_TEST1 0`
- `#define LAB13_TEST2 0`

#### 5.1.1 Macro Definition Documentation

##### 5.1.1.1 `#define LAB13_TEST1 0`

`Timer` class (Lab 13) configuration file. Activate test 'N' by defining the corresponding LAB12\_TESTN to have the value 1.

##### 5.1.1.2 `#define LAB13_TEST2 0`

### 5.2 constructor.cpp File Reference

```
#include <iostream>
#include <string>
#include "Timer.h"
#include "testvector.h"
```

#### Macros

- `#define runTest(Type) testConstructor<Type>(numValues, #Type)`

## Functions

- `template<typename DataType >`  
`int testCompute (DataType value)`
- `template<>`  
`int testCompute< int > (int value)`
- `template<>`  
`int testCompute< double > (double value)`
- `template<typename DataType >`  
`void testConstructor (int numValues, string name)`
- `int main (int argc, char **argv)`

## Variables

- `const int numRepetitions = 1000000`

### 5.2.1 Macro Definition Documentation

5.2.1.1 `#define runTest( Type ) testConstructor<Type>(numValues, #Type)`

### 5.2.2 Function Documentation

5.2.2.1 `int main ( int argc, char ** argv )`

5.2.2.2 `template<typename DataType > int testCompute ( DataType value )`

5.2.2.3 `template<> int testCompute< double > ( double value )`

5.2.2.4 `template<> int testCompute< int > ( int value )`

5.2.2.5 `template<typename DataType > void testConstructor ( int numValues, string name )`

### 5.2.3 Variable Documentation

5.2.3.1 `const int numRepetitions = 1000000`

## 5.3 inc.cpp File Reference

```
#include <iostream>
#include "Timer.h"
#include "testvector.h"
```

## Functions

- `int main (int argc, char **argv)`

## Variables

- const int [numRepetitions](#) = 1000000

### 5.3.1 Function Documentation

5.3.1.1 int main ( int *argc*, char \*\* *argv* )

### 5.3.2 Variable Documentation

5.3.2.1 const int numRepetitions = 1000000

## 5.4 search.cpp File Reference

```
#include <iostream>
#include <algorithm>
#include <vector>
#include "Timer.h"
```

## Classes

- class [Search](#)
- class [linearSearch](#)
- class [binarySearch](#)
- class [STLSearch](#)

## Functions

- int [main](#) (int argc, char \*\*argv)

## Variables

- const int [numSearches](#) = 100000

### 5.4.1 Function Documentation

5.4.1.1 int main ( int *argc*, char \*\* *argv* )

### 5.4.2 Variable Documentation

5.4.2.1 const int numSearches = 100000

## 5.5 sort.cpp File Reference

```
#include <iostream>
#include <algorithm>
#include <vector>
#include "Timer.h"
```

## Functions

- void [selectionSort](#) (vector< int >::iterator front, vector< int >::iterator back)
- void [quickSort](#) (vector< int >::iterator front, vector< int >::iterator back)
- void [timeSort](#) (void(\*fcn)(vector< int >::iterator front, vector< int >::iterator back), const string name, const vector< int > &masterList, const [Timer](#) &overhead)
- int [main](#) (int argc, char \*\*argv)

## Variables

- const int [numSorts](#) = 100

### 5.5.1 Function Documentation

5.5.1.1 int [main](#) ( int *argc*, char \*\* *argv* )

5.5.1.2 void [quickSort](#) ( vector< int >::iterator *front*, vector< int >::iterator *back* )

5.5.1.3 void [selectionSort](#) ( vector< int >::iterator *front*, vector< int >::iterator *back* )

5.5.1.4 void [timeSort](#) ( void(\*) (vector< int >::iterator front, vector< int >::iterator back) *fcn*, const string *name*, const vector< int > & *masterList*, const [Timer](#) & *overhead* )

### 5.5.2 Variable Documentation

5.5.2.1 const int [numSorts](#) = 100

## 5.6 test13.cpp File Reference

```
#include <iostream>
#include <cctype>
#include <ctime>
#include "Timer.h"
```

## Functions

- void [wait](#) (int secs)
- void [print\\_help](#) ()
- int [main](#) ()

### 5.6.1 Function Documentation

5.6.1.1 `int main ( )`

5.6.1.2 `void print_help ( )`

5.6.1.3 `void wait ( int secs )`

## 5.7 testtimer.cpp File Reference

```
#include "Timer.h"
#include <iostream>
#include <stddef.h>
#include <sys/time.h>
#include <cstdio>
```

### Functions

- double [getElapsed](#) (timeval &t1)
- int [main](#) (int argc, char \*\*argv)

### 5.7.1 Function Documentation

5.7.1.1 `double getElapsed ( timeval & t1 )`

5.7.1.2 `int main ( int argc, char ** argv )`

## 5.8 testvector.cpp File Reference

```
#include <functional>
#include <algorithm>
#include "testvector.h"
```

## 5.9 testvector.h File Reference

```
#include <stdexcept>
#include <iostream>
#include <vector>
```

### Classes

- class [TestVector](#)

## 5.10 Timer.cpp File Reference

```
#include <ctime>
#include <stdexcept>
#include <iostream>
#include <sys/time.h>
#include "Timer.h"
```

## 5.11 Timer.h File Reference

```
#include <ctime>
#include <stdexcept>
#include <iostream>
#include <sys/time.h>
```

### Classes

- class [Timer](#)

### 5.11.1 Detailed Description

#### Author

Leah Kramer

#### Date

09/26/2017 <This program="" runs="" a="" timer>="">



# Index

- binarySearch, 7
  - operator(), 7
- config.h, 13
  - LAB13\_TEST1, 13
  - LAB13\_TEST2, 13
- constructor.cpp, 13
  - main, 14
  - numRepetitions, 14
  - runTest, 14
  - testCompute, 14
  - testCompute< double >, 14
  - testCompute< int >, 14
  - testConstructor, 14
- getElapsed
  - testtimer.cpp, 17
- getElapsedTime
  - Timer, 10
- inc.cpp, 14
  - main, 15
  - numRepetitions, 15
- LAB13\_TEST1
  - config.h, 13
- LAB13\_TEST2
  - config.h, 13
- linearSearch, 7
  - operator(), 8
- main
  - constructor.cpp, 14
  - inc.cpp, 15
  - search.cpp, 15
  - sort.cpp, 16
  - test13.cpp, 17
  - testtimer.cpp, 17
- numRepetitions
  - constructor.cpp, 14
  - inc.cpp, 15
- numSearches
  - search.cpp, 15
- numSorts
  - sort.cpp, 16
- operator()
  - binarySearch, 7
  - linearSearch, 8
  - STLSearch, 9
- operator++
  - TestVector, 9
- operator[]
  - TestVector, 9
- print\_help
  - test13.cpp, 17
- quickSort
  - sort.cpp, 16
- runTest
  - constructor.cpp, 14
- STLSearch, 8
  - operator(), 9
- Search, 8
- search.cpp, 15
  - main, 15
  - numSearches, 15
- selectionSort
  - sort.cpp, 16
- sort.cpp, 15
  - main, 16
  - numSorts, 16
  - quickSort, 16
  - selectionSort, 16
  - timeSort, 16
- start
  - Timer, 10
- stop
  - Timer, 11
- test13.cpp, 16
  - main, 17
  - print\_help, 17
  - wait, 17
- testCompute
  - constructor.cpp, 14
- testCompute< double >
  - constructor.cpp, 14
- testCompute< int >
  - constructor.cpp, 14
- testConstructor
  - constructor.cpp, 14
- TestVector, 9
  - operator++, 9
  - operator[], 9
  - TestVector, 9
- testtimer.cpp, 17
  - getElapsed, 17

- main, [17](#)
- testvector.cpp, [17](#)
- testvector.h, [17](#)
- timeSort
  - sort.cpp, [16](#)
- Timer, [9](#)
  - getElapsedTime, [10](#)
  - start, [10](#)
  - stop, [11](#)
  - Timer, [10](#)
- Timer.cpp, [18](#)
- Timer.h, [18](#)
- wait
  - test13.cpp, [17](#)