

Reference Manual

Generated by Doxygen 1.6.1

Fri Nov 15 08:08:25 2013

Contents

1	Class Index	1
1.1	Class List	1
2	Class Documentation	3
2.1	Time Class Reference	3
2.1.1	Detailed Description	4
2.1.2	Constructor & Destructor Documentation	4
2.1.2.1	Time	4
2.1.2.2	Time	4
2.1.3	Member Function Documentation	4
2.1.3.1	add	4
2.1.3.2	enter	4
2.1.3.3	equal	5
2.1.3.4	getHours	5
2.1.3.5	getMinutes	5
2.1.3.6	lessThan	5
2.1.3.7	operator+=	5
2.1.3.8	operator<	6
2.1.3.9	operator==	6
2.1.3.10	print	6
2.1.3.11	println	6
2.1.3.12	read	6
2.1.3.13	setHours	7
2.1.3.14	setMinutes	7
2.1.3.15	write	7
2.1.4	Member Data Documentation	7
2.1.4.1	maxHours	7
2.1.4.2	maxMinutes	7

2.1.4.3	noon	7
---------	------	-------	---

Chapter 1

Class Index

1.1 Class List

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

Time	3
--------------------------------	-------------------

Chapter 2

Class Documentation

2.1 Time Class Reference

Public Member Functions

- `Time ()`
- `Time (const int newHours, const int newMinutes)`
- `int getHours () const`
- `int getMinutes () const`
- `void setHours (const int newHours)`
- `void setMinutes (const int newMinutes)`
- `void print (ostream &out=cout) const`
- `void println (ostream &out=cout) const`
- `void enter (istream &in=cin)`
- `void read (istream &in)`
- `void write (ostream &out) const`
- `void add (const Time &time)`
- `bool equal (const Time &time) const`
- `bool lessThan (const Time &time) const`
- `void operator+= (const Time &time)`
- `bool operator== (const Time &time) const`
- `bool operator< (const Time &time) const`

Static Public Attributes

- `static const int maxHours = 24`
- `static const int maxMinutes = 60`
- `static const Time noon`

Friends

- `ostream & operator<< (ostream &out, const Time &time)`
- `istream & operator>> (istream &in, Time &time)`

2.1.1 Detailed Description

A class for storing information about time in military format.

Author:

Mark Maloof (maloof@cs.georgetown.edu)

Version:

1.0, 8/16/13

2.1.2 Constructor & Destructor Documentation

2.1.2.1 Time::Time ()

Default constructor. Sets this time object to 00:00.

2.1.2.2 Time::Time (const int *newHours*, const int *newMinutes*)

Explicit constructor. Sets this time object to the specified hours and minutes. Out-of-range values will not replace default values.

Parameters:

newHours the hours of this time object

newMinutes the minutes of this time object

2.1.3 Member Function Documentation

2.1.3.1 void Time::add (const Time & *time*)

Adds the specified time object to this time object.

Parameters:

time the time object to be added

2.1.3.2 void Time::enter (istream & *in* = cin)

Extracts a time in the format "hh:mm" from the specified input stream. If there are problems with the format of the information in the stream, then no values are set. Out-of-range values do not replace current values.

Parameters:

in the input stream for reading

2.1.3.3 bool Time::equal (const Time & *time*) const

Returns true if this time object is equal to the specified time object; returns false otherwise.

Parameters:

time the time object to be compared

Returns:

true if the objects are equal

2.1.3.4 int Time::getHours () const

Returns the hours of this time object.

Returns:

the hours

2.1.3.5 int Time::getMinutes () const

Returns the minutes of this time object.

Returns:

the minutes

2.1.3.6 bool Time::lessThan (const Time & *time*) const

Returns true if this time object is less than the specified time object; returns false otherwise.

Parameters:

time the time object to be compared

Returns:

true if this object is less than the specified object

2.1.3.7 void Time::operator+= (const Time & *time*)

Adds the specified time object to this time object.

Parameters:

time the time object to be added

2.1.3.8 bool Time::operator< (const Time & *time*) const

Returns true if this time object is less than the specified time object; returns false otherwise.

Parameters:

time the time object to be compared

Returns:

true if this object is less than the specified object

2.1.3.9 bool Time::operator== (const Time & *time*) const

Returns true if this time object is equal to the specified time object; returns false otherwise.

Parameters:

time the time object to be compared

Returns:

true if the objects are equal

2.1.3.10 void Time::print (ostream & *out* = cout) const

Prints this time object to the specified output stream in the format "hh:mm" with no terminating end-of-line character.

Parameters:

out the output stream for printing

2.1.3.11 void Time::println (ostream & *out* = cout) const

Prints this time object to the specified output stream in the format "hh:mm" with a terminating end-of-line character.

Parameters:

out the output stream for printing

2.1.3.12 void Time::read (istream & *in*)

Reads a time in the format "hh mm" from the specified input stream. If there are problems with the format of the information in the stream, then no values are set. Out-of-range values do not replace current values.

Parameters:

in the specified input stream for reading

2.1.3.13 void Time::setHours (const int *newHours*)

Sets the hours of this object to the specified hours, which must be in {0,..., 23}. Out-of-range values do not replace current values.

Parameters:

newHours the specified hours

2.1.3.14 void Time::setMinutes (const int *newMinutes*)

Sets the minutes of this object to the specified minutes, which must be in {0,..., 59}. Out-of-range values do not replace current values.

Parameters:

newMinutes the specified minutes

2.1.3.15 void Time::write (ostream & *out*) const

Writes this time object to the specified output stream in the format "hh mm" with no terminating end-of-line character.

Parameters:

out the output stream for writing

2.1.4 Member Data Documentation

2.1.4.1 const int Time::maxHours = 24 [static]

the maximum value for hours

2.1.4.2 const int Time::maxMinutes = 60 [static]

the maximum value for minutes

2.1.4.3 const Time Time::noon [static]

a constant for noon

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

- time12.cc

Index

add
 Time, 4

enter
 Time, 4

equal
 Time, 4

getHours
 Time, 5

getMinutes
 Time, 5

lessThan
 Time, 5

maxHours
 Time, 7

maxMinutes
 Time, 7

noon
 Time, 7

operator<
 Time, 5

operator+=
 Time, 5

operator==
 Time, 6

print
 Time, 6

println
 Time, 6

read
 Time, 6

setHours
 Time, 6

setMinutes
 Time, 7

Time, 3
 add, 4
 enter, 4

equal, 4

getHours, 5

getMinutes, 5

lessThan, 5

maxHours, 7

maxMinutes, 7

noon, 7

operator<, 5

operator+=, 5

operator==, 6

print, 6

println, 6

read, 6

setHours, 6

setMinutes, 7

Time, 4

write, 7

write
 Time, 7