

labo\_02\_carvalho\_bruno\_gallay\_david

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	Date Class Reference . . . . .	5
3.1.1	Constructor & Destructor Documentation . . . . .	6
3.1.1.1	Date() [1/2] . . . . .	7
3.1.1.2	Date() [2/2] . . . . .	7
3.1.2	Member Function Documentation . . . . .	7
3.1.2.1	dayInMonth() . . . . .	7
3.1.2.2	display() . . . . .	8
3.1.2.3	getDay() . . . . .	8
3.1.2.4	getMonthEnum() . . . . .	8
3.1.2.5	getMonthNo() . . . . .	8
3.1.2.6	getMonthString() . . . . .	9
3.1.2.7	getYear() . . . . .	9
3.1.2.8	isLeap() . . . . .	9
3.1.2.9	isValid() . . . . .	9
3.1.2.10	operator std::string() . . . . .	10
3.1.2.11	operator"!="() . . . . .	10
3.1.2.12	operator++() [1/2] . . . . .	10

3.1.2.13	<a href="#">operator++()</a> [2/2]	10
3.1.2.14	<a href="#">operator+=()</a> [1/2]	11
3.1.2.15	<a href="#">operator+=()</a> [2/2]	11
3.1.2.16	<a href="#">operator-()</a>	11
3.1.2.17	<a href="#">operator--()</a> [1/2]	12
3.1.2.18	<a href="#">operator--()</a> [2/2]	12
3.1.2.19	<a href="#">operator-=()</a> [1/2]	12
3.1.2.20	<a href="#">operator-=()</a> [2/2]	12
3.1.2.21	<a href="#">operator&lt;()</a>	14
3.1.2.22	<a href="#">operator&lt;=()</a>	14
3.1.2.23	<a href="#">operator=()</a>	14
3.1.2.24	<a href="#">operator==()</a>	16
3.1.2.25	<a href="#">operator&gt;()</a>	16
3.1.2.26	<a href="#">operator&gt;=()</a>	16
3.1.2.27	<a href="#">receive()</a>	18
3.1.2.28	<a href="#">setDay()</a>	18
3.1.2.29	<a href="#">setMonth()</a> [1/3]	18
3.1.2.30	<a href="#">setMonth()</a> [2/3]	20
3.1.2.31	<a href="#">setMonth()</a> [3/3]	20
3.1.2.32	<a href="#">setYear()</a>	20
<b>4</b>	<b>File Documentation</b>	<b>23</b>
4.1	<a href="#">date.h File Reference</a>	23
4.1.1	<a href="#">Enumeration Type Documentation</a>	24
4.1.1.1	<a href="#">Month</a>	24
4.1.2	<a href="#">Function Documentation</a>	24
4.1.2.1	<a href="#">operator+()</a> [1/4]	24
4.1.2.2	<a href="#">operator+()</a> [2/4]	25
4.1.2.3	<a href="#">operator+()</a> [3/4]	25
4.1.2.4	<a href="#">operator+()</a> [4/4]	25
4.1.2.5	<a href="#">operator-()</a> [1/2]	25
4.1.2.6	<a href="#">operator-()</a> [2/2]	25
4.1.2.7	<a href="#">operator&lt;&lt;()</a>	25
4.1.2.8	<a href="#">operator&gt;&gt;()</a>	26
4.1.3	<a href="#">Variable Documentation</a>	26
4.1.3.1	<a href="#">DEFAULT_YEAR</a>	26
	<b>Index</b>	<b>27</b>

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">Date</a> . . . . .	5
--------------------------------	---



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">date.h</a> . . . . .	23
----------------------------------	----





## Chapter 3

# Class Documentation

### 3.1 Date Class Reference

```
#include <date.h>
```

#### Public Member Functions

- [Date](#) (unsigned day, unsigned month, unsigned year)  
*Create a [Date](#).*
- [Date](#) (unsigned day=1, [Month](#) month=[Month::JANUARY](#), unsigned year=[DEFAULT\\_YEAR](#))  
*Create a [Date](#).*
- [Date](#) & [setDay](#) (unsigned day)  
*Update day value.*
- [Date](#) & [setMonth](#) (unsigned month)  
*Update month value.*
- [Date](#) & [setMonth](#) ([Month](#) month)  
*Update month value.*
- [Date](#) & [setMonth](#) (std::string month)  
*Update month value.*
- [Date](#) & [setYear](#) (unsigned year)  
*Update year value.*
- unsigned [getDay](#) () const  
*get day value*
- unsigned [getMonthNo](#) () const  
*get month value*
- std::string [getMonthString](#) () const  
*get month value*
- [Month](#) [getMonthEnum](#) () const  
*get month value*
- unsigned [getYear](#) () const  
*get year value*
- bool [isValid](#) () const  
*get a bool indicating if [Date](#) values are valid*
- [operator std::string](#) ()  
*Convert month to string according to the define format.*

- `bool operator== (const Date &date) const`  
*Compare two dates.*
- `bool operator!= (const Date &date) const`  
*Compare two dates.*
- `bool operator< (const Date &date) const`  
*Compare two dates.*
- `bool operator<= (const Date &date) const`  
*Compare two dates.*
- `bool operator> (const Date &date) const`  
*Compare two dates.*
- `bool operator>= (const Date &date) const`  
*Compare two dates.*
- `Date & operator+= (unsigned days)`  
*Add X day to a date.*
- `Date & operator-= (unsigned days)`  
*Subtract X day to a date.*
- `Date & operator+= (int days)`  
*Add X day to a date.*
- `Date & operator-= (int days)`  
*Subtract X day to a date.*
- `int operator- (const Date &date) const`  
*Get the number of days between two dates.*
- `Date & operator= (const Date &date)`  
*Assign an object date to an other object date.*
- `Date & operator++ ()`  
*Pre-incrementation of an object date.*
- `Date operator++ (int)`  
*Post-incrementation of an object date.*
- `Date & operator-- ()`  
*Pre-decrementation of an object date.*
- `Date operator-- (int)`  
*Post-decrementation of an object date.*
- `std::ostream & display (std::ostream &os=std::cout) const`  
*Display date in a format DD.MM.YYYY.*
- `std::istream & receive (std::istream &is=std::cin)`  
*Receive date in a format DD.MM.YYYY.*

### Static Public Member Functions

- `static bool isLeap (unsigned year)`  
*Indicate if the year is a leap year.*
- `static unsigned dayInMonth (unsigned month, unsigned year)`  
*Get the number of day in a month.*

#### 3.1.1 Constructor & Destructor Documentation

### 3.1.1.1 Date() [1/2]

```
Date::Date (
    unsigned day,
    unsigned month,
    unsigned year )
```

Create a [Date](#).

#### Parameters

<i>day</i>	
<i>month</i>	
<i>year</i>	

### 3.1.1.2 Date() [2/2]

```
Date::Date (
    unsigned day = 1,
    Month month = Month::JANUARY,
    unsigned year = DEFAULT\_YEAR )
```

Create a [Date](#).

#### Parameters

<i>day</i>	
<i>month</i>	
<i>year</i>	

## 3.1.2 Member Function Documentation

### 3.1.2.1 dayInMonth()

```
static unsigned Date::dayInMonth (
    unsigned month,
    unsigned year ) [static]
```

Get the number of day in a month.

#### Returns

Number of day in the given month or unsigned max value

### 3.1.2.2 display()

```
std::ostream& Date::display (
    std::ostream & os = std::cout ) const
```

Display date in a format DD.MM.YYYY.

#### Parameters

<i>os</i>	
-----------	--

#### Returns

output stream "DD.MM.YYYY"

### 3.1.2.3 getDay()

```
unsigned Date::getDay ( ) const
```

get day value

#### Returns

unsigned

### 3.1.2.4 getMonthEnum()

```
Month Date::getMonthEnum ( ) const
```

get month value

#### Returns

Month enum

### 3.1.2.5 getMonthNo()

```
unsigned Date::getMonthNo ( ) const
```

get month value

#### Returns

unsigned

### 3.1.2.6 getMonthString()

```
std::string Date::getMonthString ( ) const
```

get month value

#### Returns

string

### 3.1.2.7 getYear()

```
unsigned Date::getYear ( ) const
```

get year value

#### Returns

unsigned

### 3.1.2.8 isLeap()

```
static bool Date::isLeap (
    unsigned year ) [static]
```

Indicate if the year is a leap year.

#### Returns

True if the year is leap, else return false

### 3.1.2.9 isValid()

```
bool Date::isValid ( ) const
```

get a bool indicating if [Date](#) values are valid

#### Returns

bool

### 3.1.2.10 operator std::string()

```
Date::operator std::string ( )
```

Convert month to string according to the define format.

#### Returns

Month converted into string

### 3.1.2.11 operator!=(())

```
bool Date::operator!= (
    const Date & date ) const
```

Compare two dates.

#### Parameters

<i>date</i>	
-------------	--

#### Returns

True if the two dates are inequal

### 3.1.2.12 operator++() [1/2]

```
Date& Date::operator++ ( )
```

Pre-incrementation of an object date.

#### Returns

The object date modified

### 3.1.2.13 operator++() [2/2]

```
Date Date::operator++ (
    int )
```

Post-incrementation of an object date.

#### Returns

The object date before incrementation

**3.1.2.14 operator+=( )** [1/2]

```
Date& Date::operator+= (
    unsigned days )
```

Add X day to a date.

**Parameters**

<i>days</i>	
-------------	--

**Returns**

The object date modified

**3.1.2.15 operator+=( )** [2/2]

```
Date& Date::operator+= (
    int days )
```

Add X day to a date.

**Parameters**

<i>days</i>	
-------------	--

**Returns**

The object date modified

**3.1.2.16 operator-( )**

```
int Date::operator- (
    const Date & date ) const
```

Get the number of days between two dates.

**Parameters**

<i>days</i>	
-------------	--

**Returns**

days as int

**3.1.2.17 operator--()** [1/2]

```
Date& Date::operator-- ( )
```

Pre-decrementation of an object date.

**Returns**

The object date modified

**3.1.2.18 operator--()** [2/2]

```
Date Date::operator-- (
    int )
```

Post-decrementation of an object date.

**Returns**

The object date before decrementation

**3.1.2.19 operator-=()** [1/2]

```
Date& Date::operator-= (
    unsigned days )
```

Substract X day to a date.

**Parameters**

<i>days</i>	
-------------	--

**Returns**

The object date modified

**3.1.2.20 operator-=()** [2/2]

```
Date& Date::operator-= (
    int days )
```



Substract X day to a date.

**Parameters**

<i>days</i>	
-------------	--

**Returns**

The object date modified

**3.1.2.21 operator<()**

```
bool Date::operator< (
    const Date & date ) const
```

Compare two dates.

**Parameters**

<i>date</i>	
-------------	--

**Returns**

True if the date tested is lower than the param date

**3.1.2.22 operator<=()**

```
bool Date::operator<= (
    const Date & date ) const
```

Compare two dates.

**Parameters**

<i>date</i>	
-------------	--

**Returns**

True if the date tested is lower or equal to the param date

**3.1.2.23 operator=()**

```
Date& Date::operator= (
    const Date & date )
```

Assign an object date to an other object date.

**Parameters**

<i>date</i>	
-------------	--

**Returns**

The object date modified

**3.1.2.24 operator==()**

```
bool Date::operator== (
    const Date & date ) const
```

Compare two dates.

**Parameters**

<i>date</i>	
-------------	--

**Returns**

True if the two dates are equal

**3.1.2.25 operator>()**

```
bool Date::operator> (
    const Date & date ) const
```

Compare two dates.

**Parameters**

<i>date</i>	
-------------	--

**Returns**

True if the date tested is higher than the param date

**3.1.2.26 operator>=()**

```
bool Date::operator>= (
    const Date & date ) const
```

Compare two dates.

**Parameters**

<i>date</i>	
-------------	--

**Returns**

True if the date tested is higher or equal to the param date

**3.1.2.27 receive()**

```
std::istream& Date::receive (
    std::istream & is = std::cin )
```

Receive date in a format DD.MM.YYYY.

**Parameters**

<i>is</i>	
-----------	--

**Returns**

input stream "DD.MM.YYYY"

**3.1.2.28 setDay()**

```
Date& Date::setDay (
    unsigned day )
```

Update day value.

**Parameters**

<i>day</i>	
------------	--

**Returns**

Reference on this

**3.1.2.29 setMonth()** [1/3]

```
Date& Date::setMonth (
    unsigned month )
```

Update month value.

**Parameters**

<i>month</i>	
--------------	--

**Returns**

Reference on this

**3.1.2.30 setMonth()** [2/3]

```
Date& Date::setMonth (
    Month month )
```

Update month value.

**Parameters**

<i>month</i>	
--------------	--

**Returns**

Reference on this

**3.1.2.31 setMonth()** [3/3]

```
Date& Date::setMonth (
    std::string month )
```

Update month value.

**Parameters**

<i>month</i>	
--------------	--

**Returns**

Reference on this

**3.1.2.32 setYear()**

```
Date& Date::setYear (
    unsigned year )
```



Update year value.

**Parameters**

<i>year</i>	
-------------	--

**Returns**

Reference on this

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

- [date.h](#)

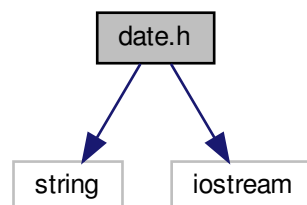
## Chapter 4

# File Documentation

### 4.1 date.h File Reference

```
#include <string>
#include <iostream>
```

Include dependency graph for date.h:



#### Classes

- class [Date](#)

#### Enumerations

- enum [Month](#) {  
    [Month::JANUARY](#) = 1, [Month::FEBRUARY](#), [Month::MARCH](#), [Month::APRIL](#),  
    [Month::MAY](#), [Month::JUNE](#), [Month::JULY](#), [Month::AUGUST](#),  
    [Month::SEPTEMBER](#), [Month::OCTOBER](#), [Month::NOVEMBER](#), [Month::DECEMBER](#) }

## Functions

- `std::ostream & operator<< (std::ostream &os, const Date &date)`
- `std::istream & operator>> (std::istream &is, Date &date)`
- `Date operator+ (Date date, unsigned days)`
- `Date operator+ (unsigned days, const Date date)`
- `Date operator- (Date date, unsigned days)`
- `Date operator+ (Date date, int days)`
- `Date operator+ (int days, const Date &date)`
- `Date operator- (Date date, int days)`

## Variables

- `const unsigned DEFAULT_YEAR = 1970`

### 4.1.1 Enumeration Type Documentation

#### 4.1.1.1 Month

```
enum Month [strong]
```

##### Enumerator

JANUARY	
FEBRUARY	
MARCH	
APRIL	
MAY	
JUNE	
JULY	
AUGUST	
SEPTEMBER	
OCTOBER	
NOVEMBER	
DECEMBER	

### 4.1.2 Function Documentation

#### 4.1.2.1 operator+() [1/4]

```
Date operator+ (
    Date date,
    unsigned days )
```

#### 4.1.2.2 operator+() [2/4]

```
Date operator+ (
    unsigned days,
    const Date date )
```

#### 4.1.2.3 operator+() [3/4]

```
Date operator+ (
    Date date,
    int days )
```

#### 4.1.2.4 operator+() [4/4]

```
Date operator+ (
    int days,
    const Date & date )
```

#### 4.1.2.5 operator-() [1/2]

```
Date operator- (
    Date date,
    unsigned days )
```

#### 4.1.2.6 operator-() [2/2]

```
Date operator- (
    Date date,
    int days )
```

#### 4.1.2.7 operator<<()

```
std::ostream& operator<< (
    std::ostream & os,
    const Date & date )
```

#### 4.1.2.8 operator>>()

```
std::istream& operator>> (
    std::istream & is,
    Date & date )
```

### 4.1.3 Variable Documentation

#### 4.1.3.1 DEFAULT\_YEAR

```
const unsigned DEFAULT_YEAR = 1970
```

# Index

DEFAULT\_YEAR

date.h, 26

Date, 5

Date, 6, 7

dayInMonth, 7

display, 7

getDay, 8

getMonthEnum, 8

getMonthNo, 8

getMonthString, 8

getYear, 9

isLeap, 9

isValid, 9

operator std::string, 9

operator!=, 10

operator<, 14

operator<=, 14

operator>, 16

operator>=, 16

operator++, 10

operator+=, 10, 11

operator-, 11

operator--, 12

operator-=, 12

operator=, 14

operator==, 16

receive, 18

setDay, 18

setMonth, 18, 20

setYear, 20

date.h, 23

DEFAULT\_YEAR, 26

Month, 24

operator<<, 25

operator>>, 25

operator+, 24, 25

operator-, 25

dayInMonth

Date, 7

display

Date, 7

getDay

Date, 8

getMonthEnum

Date, 8

getMonthNo

Date, 8

getMonthString

Date, 8

getYear

Date, 9

isLeap

Date, 9

isValid

Date, 9

Month

date.h, 24

operator std::string

Date, 9

operator!=

Date, 10

operator<

Date, 14

operator<<

date.h, 25

operator<=

Date, 14

operator>

Date, 16

operator>>

date.h, 25

operator>=

Date, 16

operator+

date.h, 24, 25

operator++

Date, 10

operator+=

Date, 10, 11

operator-

Date, 11

date.h, 25

operator--

Date, 12

operator-=

Date, 12

operator=

Date, 14

operator==

Date, 16

receive

Date, 18

setDay

Date, 18

setMonth

Date, [18](#), [20](#)  
setYear  
    Date, [20](#)