

## CLASE CUBO

Generado por Doxygen 1.9.8



# Chapter 1

## Índice de clases

### 1.1 Lista de clases

Lista de clases, estructuras, uniones e interfaces con breves descripciones:

<a href="#">Cube</a>	Representa un cubo con una longitud de arista determinada . . . . .	??
----------------------	---	----



## Chapter 2

# Índice de archivos

### 2.1 Lista de archivos

Lista de todos los archivos con breves descripciones:

<a href="#">cubo.cpp</a>	.....	??
--------------------------	-------	----



## Chapter 3

# Documentación de clases

### 3.1 Referencia de la clase Cube

Representa un cubo con una longitud de arista determinada.

Diagrama de colaboración de Cube:

Cube
- length
+ Cube()
+ Cube()
+ getLength()
+ setLength()
+ getVolume()
+ getSurfaceArea()

#### Métodos públicos

- [Cube](#) (double l=1.0)  
*Constructor con parámetro.*
- [Cube](#) ()  
*Constructor por defecto.*
- double [getLength](#) () const  
*Obtiene la longitud actual de la arista del cubo.*
- void [setLength](#) (double l)  
*Establece una nueva longitud para la arista del cubo.*
- double [getVolume](#) () const  
*Calcula el volumen del cubo.*
- double [getSurfaceArea](#) () const  
*Calcula el área superficial del cubo.*

### Atributos privados

- double `length`

*Longitud de la arista del cubo.*

### 3.1.1 Descripción detallada

Representa un cubo con una longitud de arista determinada.

La clase incluye métodos para obtener y modificar la longitud del cubo, así como para calcular su volumen y área superficial.

#### Nota

Se puede ampliar fácilmente para representar prismas rectangulares.

#### Atención

No acepta longitudes negativas ni nulas.

#### Ver también

[getVolume\(\)](#), [getSurfaceArea\(\)](#), [setLength\(\)](#)

### 3.1.2 Documentación de constructores y destructores

#### 3.1.2.1 `Cube()` [1/2]

```
Cube::Cube (
    double l = 1.0 ) [inline]
```

Constructor con parámetro.

##### Parámetros

/	Longitud inicial de la arista (por defecto 1.0).
---	--

#### 3.1.2.2 `Cube()` [2/2]

```
Cube::Cube ( ) [inline]
```

Constructor por defecto.

Inicializa el cubo con una longitud de 1.0.



### 3.1.3 Documentación de funciones miembro

#### 3.1.3.1 getLength()

```
double Cube::getLength ( ) const [inline]
```

Obtiene la longitud actual de la arista del cubo.

**Devuelve**

Longitud de la arista.

Gráfico de llamadas a esta función:



#### 3.1.3.2 getSurfaceArea()

```
double Cube::getSurfaceArea ( ) const [inline]
```

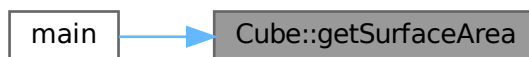
Calcula el área superficial del cubo.

Fórmula:  $A = 6 \times l^2$

**Devuelve**

Área superficial del cubo.

Gráfico de llamadas a esta función:



### 3.1.3.3 `getVolume()`

```
double Cube::getVolume ( ) const [inline]
```

Calcula el volumen del cubo.

Fórmula:  $V = l^3$

**Devuelve**

Volumen del cubo.

Gráfico de llamadas a esta función:



### 3.1.3.4 `setLength()`

```
void Cube::setLength (
    double l ) [inline]
```

Establece una nueva longitud para la arista del cubo.

Solo se aceptan valores positivos.

**Parámetros**

<code>l</code>	Nueva longitud de la arista.
----------------	------------------------------

Ver también

[getLength\(\)](#)

Gráfico de llamadas a esta función:



### 3.1.4 Documentación de datos miembro

#### 3.1.4.1 length

```
double Cube::length [private]
```

Longitud de la arista del cubo.

La documentación de esta clase está generada del siguiente archivo:

- [cubo.cpp](#)



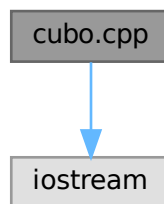
## Chapter 4

# Documentación de archivos

### 4.1 Referencia del archivo cubo.cpp

```
#include <iostream>
```

Gráfico de dependencias incluidas en cubo.cpp:



#### Clases

- class [Cube](#)  
*Representa un cubo con una longitud de arista determinada.*

#### Funciones

- int [main](#) (int argc, char \*\*argv)  
*Función principal del programa.*

#### 4.1.1 Documentación de funciones

##### 4.1.1.1 main()

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

Función principal del programa.

Crea objetos [Cube](#), muestra sus propiedades y modifica sus longitudes.

**Parámetros**

<i>argc</i>	Número de argumentos.
<i>argv</i>	Vector con los argumentos.

**Devuelve**

0 si el programa finaliza correctamente.

< Crea un cubo con longitud 3.

< Crea un cubo con longitud por defecto (1). Gráfico de llamadas de esta función:

