

C++classesDemonstratedthroughspheres

1

Generated by Doxygen 1.8.18

<b>1 Class Index</b>	<b>1</b>
<b>1 Class Index</b>	<b>1</b>
1.1 Class List . . . . .	1
<b>2 File Index</b>	<b>1</b>
2.1 File List . . . . .	1
<b>3 Class Documentation</b>	<b>2</b>
3.1 Sphere Class Reference . . . . .	2
3.1.1 Constructor & Destructor Documentation . . . . .	2
3.1.2 Member Function Documentation . . . . .	3
3.1.3 Member Data Documentation . . . . .	4
3.2 sphere Class Reference . . . . .	5
3.2.1 Detailed Description . . . . .	5
<b>4 File Documentation</b>	<b>5</b>
4.1 driver.cpp File Reference . . . . .	5
4.1.1 Function Documentation . . . . .	5
4.2 sphere.cpp File Reference . . . . .	5
4.2.1 Macro Definition Documentation . . . . .	6
4.3 sphere.h File Reference . . . . .	6
<b>Index</b>	<b>7</b>

# 1 Class Index

## 1.1 Class List

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

<b>Sphere</b>	<b>2</b>
<b>sphere</b>	
Class to manage spheres This class has an overload function that allows a sphere to be passed, and then reads out the values for that sphere like radius, diameter and volume	5

## 2 File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<b>driver.cpp</b>	<b>5</b>
<b>sphere.cpp</b>	<b>5</b>
<b>sphere.h</b>	<b>6</b>

## 3 Class Documentation

### 3.1 Sphere Class Reference

```
#include <sphere.h>
```

#### Public Member Functions

- **Sphere** ()  
*Constructs a new instance.*
- **Sphere** (double)  
*Constructs a new instance.*
- double **getRadius** () const  
*Gets the radius.*
- double **getDiameter** () const  
*Gets the diameter.*
- double **getCircumference** () const  
*Gets the circumference.*
- double **getArea** () const  
*Gets the area.*
- double **getVolume** () const  
*Gets the volume.*
- void **setRadius** (double)  
*Sets the radius.*
- std::ostream & **display** (std::ostream &)  
*Displays the given out.*

#### Private Attributes

- double **radius**

#### 3.1.1 Constructor & Destructor Documentation

##### 3.1.1.1 **Sphere()** [1/2] `Sphere::Sphere ( )`

Constructs a new instance.

##### 3.1.1.2 **Sphere()** [2/2] `Sphere::Sphere ( double r )`

Constructs a new instance.

## Parameters

<i>in</i>	<i>r</i>	radius of sphere being created
-----------	----------	--------------------------------

## 3.1.2 Member Function Documentation

**3.1.2.1 display()** `std::ostream & Sphere::display (std::ostream & out )`

Displays the given out.

## Parameters

<i>out</i>	The output to pass along
------------	--------------------------

## Returns

an ostream item to be chained/linked to cout for passing to stdio

**3.1.2.2 getArea()** `double Sphere::getArea ( ) const`

Gets the area.

## Returns

The area.

**3.1.2.3 getCircumference()** `double Sphere::getCircumference ( ) const`

Gets the circumference.

## Returns

The circumference.

**3.1.2.4 getDiameter()** `double Sphere::getDiameter ( ) const`

Gets the diameter.

**Returns**

The diameter.

**3.1.2.5 getRadius()** `double Sphere::getRadius ( ) const`

Gets the radius.

**Returns**

The radius.

**3.1.2.6 getVolume()** `double Sphere::getVolume ( ) const`

Gets the volume.

**Returns**

The volume.

**3.1.2.7 setRadius()** `void Sphere::setRadius (`  
`double r )`

Sets the radius.

**Parameters**

in	<i>r</i>	The new value to change the radius to.
----	----------	--

**3.1.3 Member Data Documentation****3.1.3.1 radius** `double Sphere::radius [private]`

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

- **sphere.h**
- **sphere.cpp**

## 3.2 sphere Class Reference

Class to manage spheres This class has an overload function that allows a sphere to be passed, and then reads out the values for that sphere like radius, diameter and volume.

```
#include <sphere.h>
```

### 3.2.1 Detailed Description

Class to manage spheres This class has an overload function that allows a sphere to be passed, and then reads out the values for that sphere like radius, diameter and volume.

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

- **sphere.h**

## 4 File Documentation

### 4.1 driver.cpp File Reference

```
#include "sphere.h"
```

#### Functions

- `std::ostream & operator<< (std::ostream &strm, Sphere &a)`
- `int main ()`

#### 4.1.1 Function Documentation

**4.1.1.1 main()** `int main ( )`

**4.1.1.2 operator<<()** `std::ostream& operator<< (`  
`std::ostream & strm,`  
`Sphere & a )`

### 4.2 sphere.cpp File Reference

```
#include "sphere.h"
```

## Macros

- `#define M_PI 3.14159265`  
*created to add more precision to pi for calculations*

### 4.2.1 Macro Definition Documentation

#### 4.2.1.1 M\_PI `#define M_PI 3.14159265`

created to add more precision to pi for calculations

## 4.3 sphere.h File Reference

```
#include <iostream>
```

## Classes

- class **Sphere**

## Index

- display
  - Sphere, 3
- driver.cpp, 5
  - main, 5
  - operator<<, 5
- getArea
  - Sphere, 3
- getCircumference
  - Sphere, 3
- getDiameter
  - Sphere, 3
- getRadius
  - Sphere, 4
- getVolume
  - Sphere, 4
- M\_PI
  - sphere.cpp, 6
- main
  - driver.cpp, 5
- operator<<
  - driver.cpp, 5
- radius
  - Sphere, 4
- setRadius
  - Sphere, 4
- Sphere, 2
  - display, 3
  - getArea, 3
  - getCircumference, 3
  - getDiameter, 3
  - getRadius, 4
  - getVolume, 4
  - radius, 4
  - setRadius, 4
  - Sphere, 2
- sphere, 5
- sphere.cpp, 5
  - M\_PI, 6
- sphere.h, 6