Reference Manual

Generated by Doxygen 1.8.13

Contents

1 Hierarchical Index

1.1 Class Hierarchy

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

2 Class Index

2.1 Class List

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

3 File Index

3.1 File List

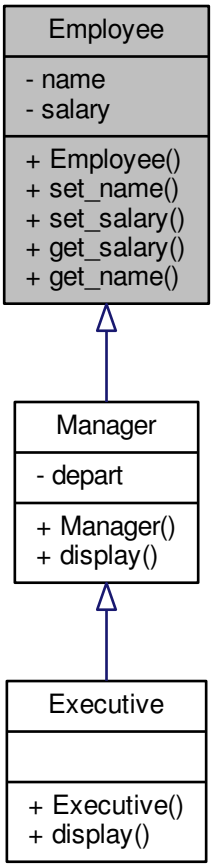
Here is a list of all files with brief descriptions:

4 Class Documentation

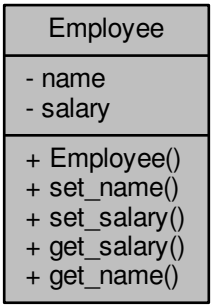
4.1 Employee Class Reference

`#include <Employee.h>`

Inheritance diagram for Employee:



Collaboration diagram for Employee:



Public Member Functions

- Employee ()
- void set_name (std::string n)
- void set_salary (double s)
- double get_salary () const
- std::string get_name () const

Private Attributes

- std::string name
- double salary

4.1.1 Detailed Description

This is a base class from which we will derive a Manager class.

Definition at line 8 of file Employee.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Employee()

```
Employee::Employee ( )
```

Default constructor which has an empty name and salary of 0.0

Definition at line 2 of file Employee.cpp.

```
3       :salary(0),name("")
4 {}
```

4.1.3 Member Function Documentation

4.1.3.1 get_name()

```
std::string Employee::get_name ( ) const
```

Function returns name

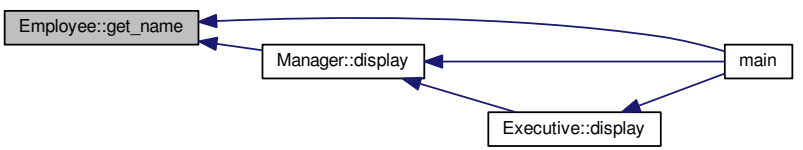
Returns

returns name of Employee

Definition at line 17 of file Employee.cpp.

```
18 {
19         return name;
20 }
```

Here is the caller graph for this function:



4.1.3.2 get_salary()

```
double Employee::get_salary ( ) const
```

Function returns salary

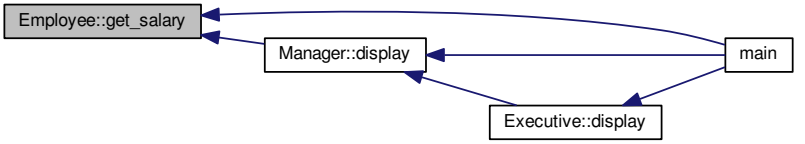
Returns

returns salary of Employee

Definition at line 5 of file Employee.cpp.

```
6 {
7         return salary;
8 }
```

Here is the caller graph for this function:



4.1.3.3 set_name()

```
void Employee::set_name (
            std::string n )
```

Function which takes a string and sets n to name

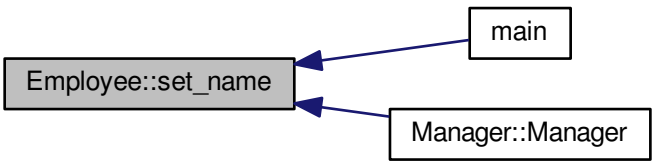
Parameters

n	the name of Employee

Definition at line 13 of file Employee.cpp.

```
14 {
15         name = n;
16 }
```

Here is the caller graph for this function:



4.1.3.4 set_salary()

```
void Employee::set_salary (
            double s )
```

Function which takes a double and sets s to salary

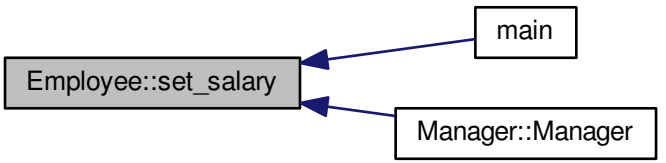
Parameters

s	the salary of Employee

Definition at line 9 of file Employee.cpp.

```
10 {
11         salary = d;
12 }
```

Here is the caller graph for this function:



4.1.4 Member Data Documentation

4.1.4.1 name

```
std::string Employee::name  [private]
```

Definition at line 37 of file Employee.h.

4.1.4.2 salary

```
double Employee::salary  [private]
```

Definition at line 38 of file Employee.h.

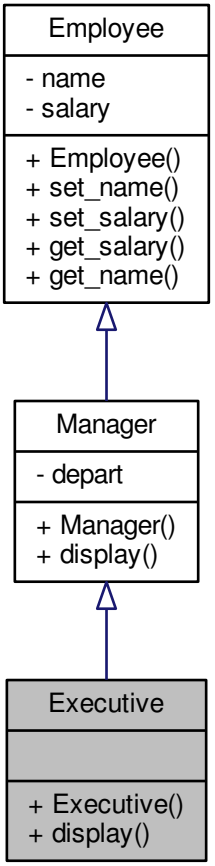
The documentation for this class was generated from the following files:

- Employee.h
- Employee.cpp

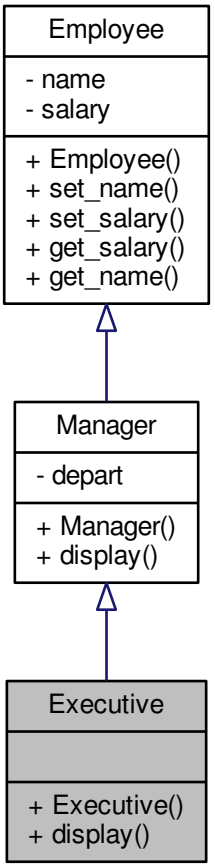
4.2 Executive Class Reference

```
#include <Executive.h>
```

Inheritance diagram for Executive:



Collaboration diagram for Executive:



Public Member Functions

- Executive (std::string d, std::string n, double s)
- void display () const

4.2.1 Detailed Description

Inherit from Manager class Add member function display to print the department, name and salary w/ Executive

Definition at line 10 of file Executive.h.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Executive()

```
Executive::Executive (
            std::string d,
            std::string n,
            double s )
```

Executive Constructor which fills in the department, name, and the salary into Executive

Parameters

d	the department which the Executive manages
n	the name of the Executive
s	the salary of the Executive

Definition at line 2 of file Executive.cpp.

```
3       :Manager(d,n,s)
4 {}
```

4.2.3 Member Function Documentation

4.2.3.1 display()

```
void Executive::display ( ) const
```

assume this will display to terminal: Displays name, department, salary and has Executive before name

Definition at line 5 of file Executive.cpp.

```
6 {
7     std::cout << "Executive ";
8     Manager::display();
9 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

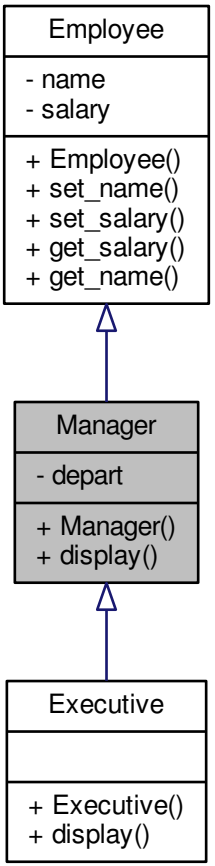
The documentation for this class was generated from the following files:

- Executive.h
- Executive.cpp

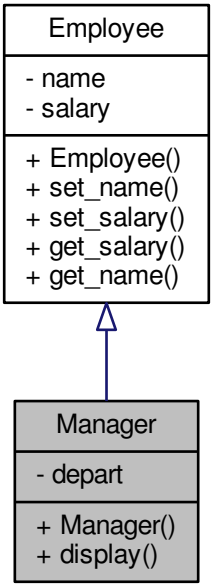
4.3 Manager Class Reference

```
#include <Manager.h>
```

Inheritance diagram for Manager:



Collaboration diagram for Manager:



Public Member Functions

- Manager (std::string d, std::string n, double s)
- void display () const

Private Attributes

- std::string depart

4.3.1 Detailed Description

Inherit from Employee class and add data member to store name of department Add member function display to print the department, name and salary

Definition at line 11 of file Manager.h.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 Manager()

```
Manager::Manager (
            std::string d,
            std::string n,
            double s )
```

Manager Constructor which fills in the department, name, and the salary into Manager

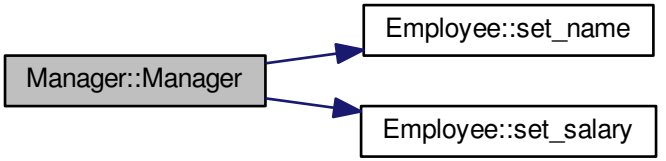
Parameters

d	the department which the Manager manages
n	the name of the Manager
s	the salary of the Manager

Definition at line 2 of file Manager.cpp.

```
3       :depart(d)
4  {
5       set_name(n);
6       set_salary(s);
7  }
```

Here is the call graph for this function:



4.3.3 Member Function Documentation

4.3.3.1 display()

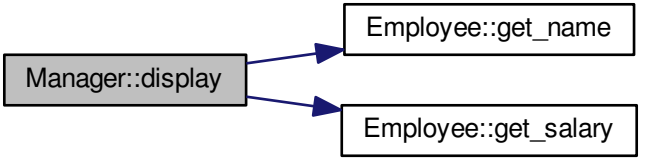
```
void Manager::display ( ) const
```

assume this will display to terminal: Function that displays name, department, and salary of Manager - uses get_salary() and get_name() from Employee

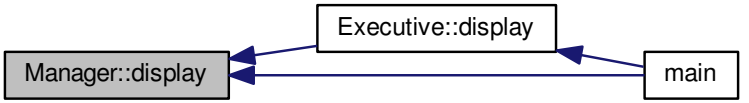
Definition at line 8 of file Manager.cpp.

```
9 {
10      std::cout << get_name()
11      << " manages department :"<< depart << std::endl
12      << "and makes: $" << get_salary() << std::endl;
13 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



4.3.4 Member Data Documentation

4.3.4.1 depart

```
std::string Manager::depart  [private]
```

Definition at line 30 of file Manager.h.

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

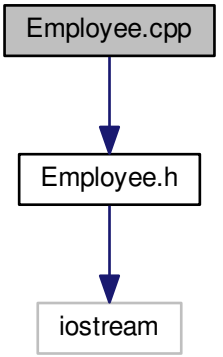
- Manager.h
- Manager.cpp

5 File Documentation

5.1 Employee.cpp File Reference

```
#include "Employee.h"
```

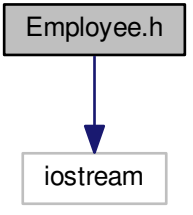
Include dependency graph for Employee.cpp:



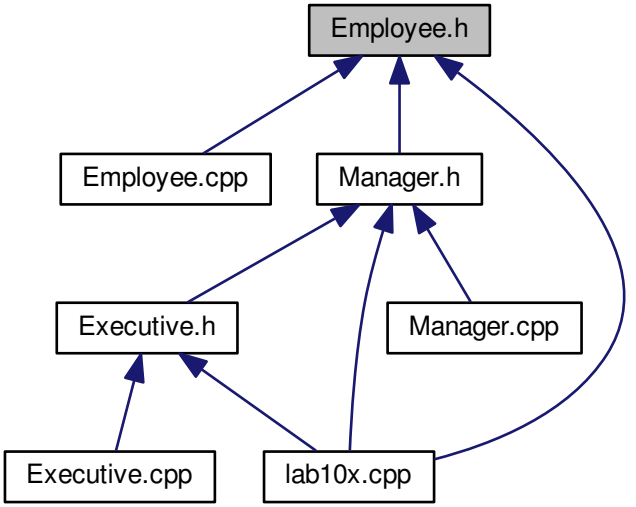
5.2 Employee.h File Reference

```
#include <iostream>
```

Include dependency graph for Employee.h:



This graph shows which files directly or indirectly include this file:



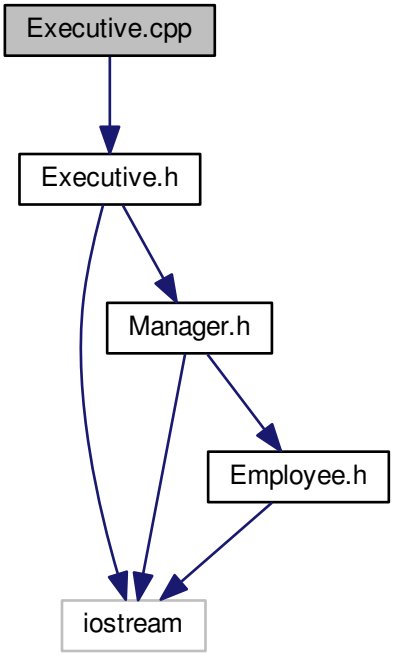
Classes

- class Employee

5.3 Executive.cpp File Reference

```
#include "Executive.h"
```

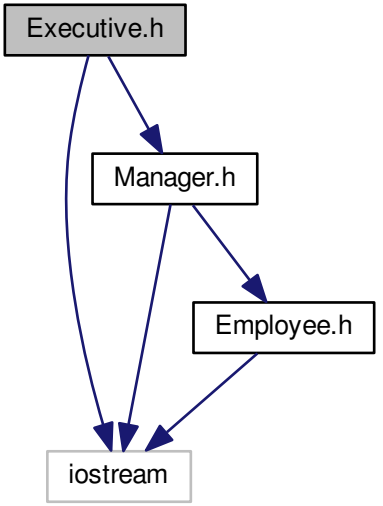
Include dependency graph for Executive.cpp:



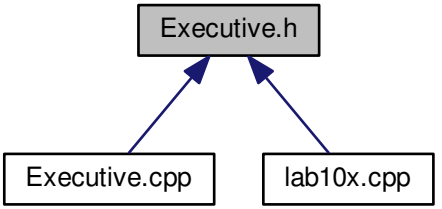
5.4 Executive.h File Reference

```
#include <iostream>
#include "Manager.h"
```

Include dependency graph for Executive.h:



This graph shows which files directly or indirectly include this file:



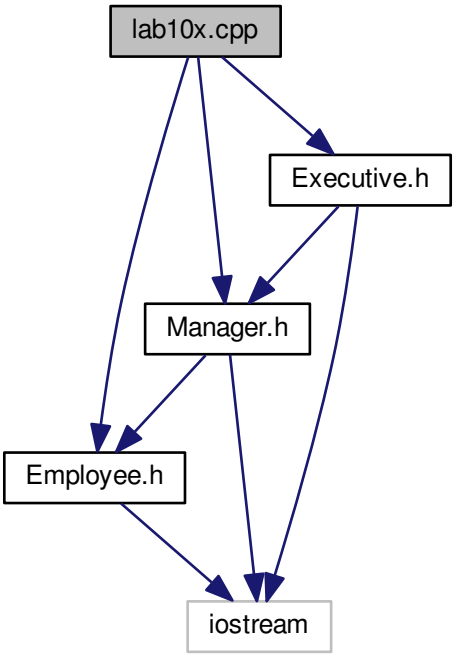
Classes

- class Executive

5.5 lab10x.cpp File Reference

```
#include "Employee.h"
#include "Manager.h"
#include "Executive.h"
```

Include dependency graph for lab10x.cpp:



Functions

- int main ()

5.5.1 Function Documentation

5.5.1.1 main()

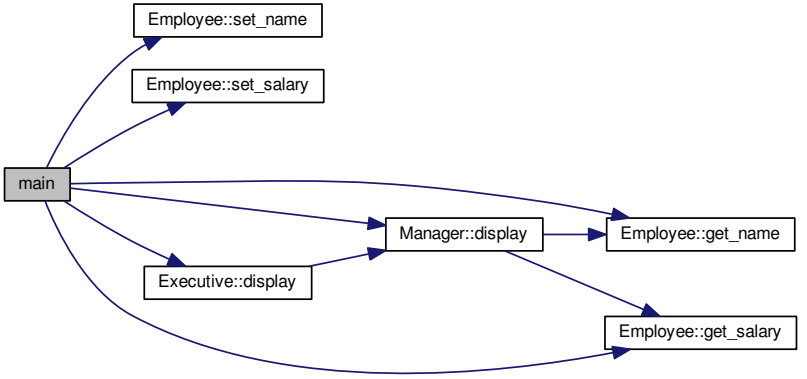
```
int main ( )
```

This creates an Employee e, Manager m, and an Executive ex and outputs their corresponding features.

Definition at line 9 of file lab10x.cpp.

```
10 {
11     Employee e;
12     e.set_name("Bob");
13     e.set_salary(10);
14     std::cout << e.get_name() << " ";
15     std::cout << e.get_salary() << std::endl;
16     Manager m("science","Bob",100);
17     m.display();
18     Executive ex("hiring","Joe",1000);
19     ex.display();
20 }
```

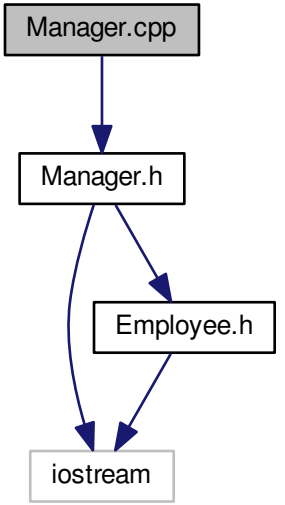
Here is the call graph for this function:



5.6 Manager.cpp File Reference

```
#include "Manager.h"
```

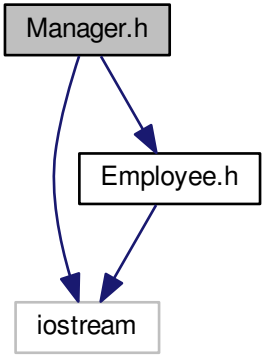
Include dependency graph for Manager.cpp:



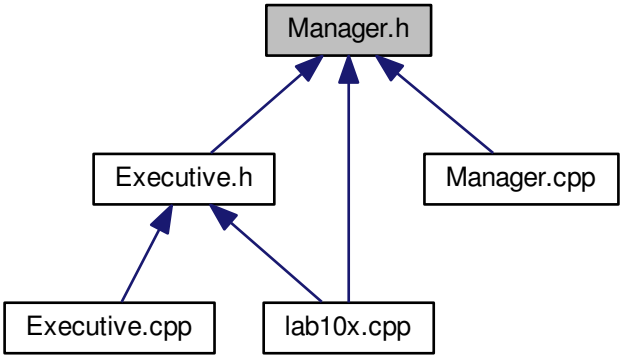
5.7 Manager.h File Reference

```
#include <iostream>
#include "Employee.h"
```

Include dependency graph for Manager.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Manager

Index