A School Directory Application

Generated by Doxygen 1.9.6

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

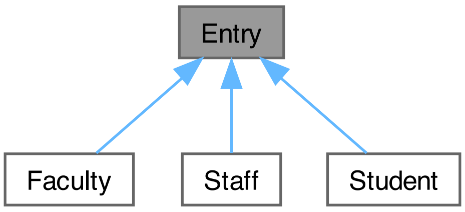
Chapter 4

Class Documentation

4.1 Entry Class Reference

```
#include <Entry.h>
```

Inheritance diagram for Entry:



Public Member Functions

- Entry ()
- Entry (std::string first, std::string last, std::string eAddress)
- bool equals (std::string first, std::string second)
- bool equals (Entry otherEntry)
- bool comesBefore (std::string first, std::string second)
- bool comesBefore (Entry otherEntry)
- virtual std::ostream & print (std::ostream &os) const
- virtual ∼Entry ()

Friends

- std::ostream & operator<< (std::ostream &os, const Entry &ent)

4.1.1 Detailed Description

Remarks

Entry: a base class for entries within a School Directory ∗ A base entry holds a first name, last name, and email address ∗

•

Base capabilities include: ∗ Base constructors ∗ Comparison operations depend upon last names, then first names ∗ Formatted output ∗

•

: files include header (Entry.h) and Implementation (Entry.cpp) ∗

•

Uncomment a main program for unit testing ∗

•

Author

Henry M. Walker ∗

Date

January 11, 2023 ∗

•

Remarks

References ∗

A School Directory as an Example of Object-Oriented Design ∗ `http://localhost/courses/cpp-style-guide/c` `php` ∗

•

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Entry() [1/2]

`Entry::Entry ( )`

Remarks

Default constructor (with no parameters) ∗

Remarks

Entry: a base class for entries within a School Directory ∗ A base entry holds a first name, last name, and email address ∗

•

Base capabilities include: ∗ Base constructors ∗ Comparison operations depend upon last names, then first names ∗ Formatted output ∗

•

: files include header (Entry.h) and Implementation (Entry.cpp) ∗

•

Uncomment a main program for unit testing ∗

•

Author

Henry M. Walker ∗

Date

January 11, 2023 ∗

-

Remarks

References ∗

A School Directory as an Example of Object-Oriented Design ∗ http://localhost/courses/cpp-style-guide/c
php ∗

-

Default constructor (with no parameters) ∗

4.1.2.2 Entry() [2/2]

```
Entry::Entry (
            std::string first,
            std::string last,
            std::string eAddress )
```

Remarks

Full-parameter constructor ∗

-

Parameters

first	a person's first name ∗
last	a person's last name ∗
eAddress	a person's email address ∗ •

4.1.2.3 ∼Entry()

```
Entry::∼Entry ( )  [virtual]
```

Remarks

a Default destructor ∗

identify as virtual, since Entry has virtual functions ∗

Remarks

a Default descructor ∗

4.1.3 Member Function Documentation

4.1.3.1 comesBefore() `[1/2]`

```
bool Entry::comesBefore (
             Entry otherEntry )
```

Remarks

check if this object comes before the parameter object ∗

-

Parameters

otherEntry	an entry to be compared with this object ∗ •

Returns

true if Entry's first/last names come before parameter's names ∗ in directory order ∗

4.1.3.2 comesBefore() `[2/2]`

```
bool Entry::comesBefore (
             std::string first,
             std::string second )
```

Remarks

check if this object comes before the given first/last names ∗

-

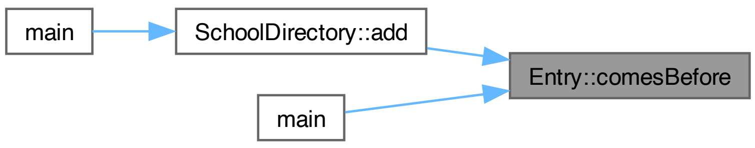
Parameters

first	a person's first name ∗
last	a person's last name ∗ •

Returns

true if Entry's first/last names come before parameter names ∗ in directory order ∗

Here is the caller graph for this function:



4.1.3.3 equals() [1/2]

```
bool Entry::equals (
            Entry otherEntry )
```

Remarks

- check whether first and last names or two Entries match ∗

Parameters

otherEntry	an entry to be compared with this object ∗

-

Returns

- true if this Entr'sy names match those of the parameter ∗

Remarks

check if this object comes before the given first/last names ∗

Parameters

first	a person's first name ∗
last	a person's last name ∗

-

Returns

- true if Entry's first/last names come before parameter names ∗ in directory order ∗

4.1.3.4 equals() [2/2]

```
bool Entry::equals (
            std::string first,
            std::string second )
```

Remarks

check whether first and last name of an Entry match two strings∗

-

Parameters

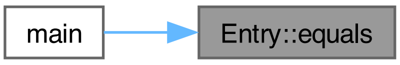
first	a person's first name ∗
last	a person's last name ∗ •

Returns

true if Entry names match first and last name strings ∗

-

Here is the caller graph for this function:



4.1.3.5 print()

```
std::ostream & Entry::print (
              std::ostream & os ) const  [virtual]
```

Remarks

output a format Entry object ∗

-

Parameters

os	output stream which will receive the formatted Entry data ∗ • @ewmrk by being a virtual function, implementations in subclasses ∗ will be interpreted via polymorphism ∗ •

Returns

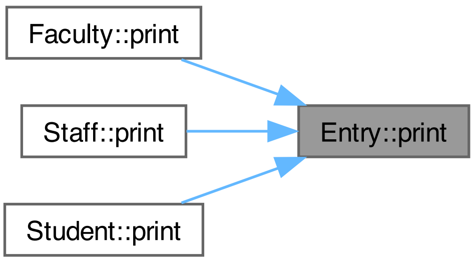
formatted string on the given output stream ∗

-

Remarks

> use of a to_string function with a string return type ∗ would require allocating space for a long string, ∗ yielding a potential memory leak ∗
>
> -

Reimplemented in Faculty, Staff, and Student.
Here is the caller graph for this function:



4.1.4 Friends And Related Function Documentation

4.1.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const Entry & ent )  [friend]
```

Remarks

> overload << for printing an Entry ∗
>
> -
>
> use of the virtual print method allows tailored output ∗ by subclasses ∗
>
> -

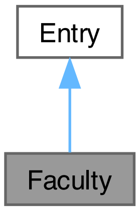
The documentation for this class was generated from the following files:

- Entry.h
- Entry.cpp

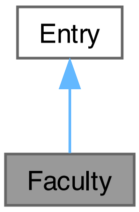
4.2 Faculty Class Reference

```
#include <Faculty.h>
```

Inheritance diagram for Faculty:



Collaboration diagram for Faculty:



Public Member Functions

- Faculty (std::string first, std::string last, std::string addr, std::string room, int ext, std::string department, int yr)
- virtual std::ostream & print (std::ostream &os) const

Public Member Functions inherited from Entry

- Entry ()
- Entry (std::string first, std::string last, std::string eAddress)
- bool equals (std::string first, std::string second)
- bool equals (Entry otherEntry)
- bool comesBefore (std::string first, std::string second)
- bool comesBefore (Entry otherEntry)
- virtual std::ostream & print (std::ostream &os) const
- virtual ∼Entry ()

4.2.1 Detailed Description

Remarks

Faculty: a class derived from Entry for a School Directory ∗ faculty entry ∗ inherits a first name, last name, and email address from Entry ∗ additional fields are the faculty member's office, extension, ∗ department, and year of initial aappointment to the school ∗

-

Inherited capabilities include: ∗ Base constructors ∗ Comparison operations depend upon last names, then first names ∗ Formatted orubt abd output ∗

-

Overwritten capabilities include: ∗ Multi-parameter constructor ∗ Formatted print method ∗

-

: files include header (Faculty.h), Implementation (Faculty.cpp)∗

-

Uncomment a main program for unit testing ∗

-

Author

Henry M. Walker ∗

Date

January 11, 2023 ∗

-

Remarks

References ∗

A School Directory as an Example of Object-Oriented Design ∗ http://localhost/courses/cpp-style-guide/d
php ∗

-

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Faculty()

```
Faculty::Faculty (
              std::string first,
              std::string last,
              std::string addr,
              std::string room,
              int ext,
              std::string department,
              int yr )
```

Remarks

Full-parameter constructor ∗

-

Parameters

first	a faculty member's first name ∗
last	a faculty member's last name ∗
eAddress	a faculty member's email address ∗
room	a faculty member's office ∗
ext	the telephone number extension for the office ∗
department	the faculty member's [primary] department ∗
yr	the year of the faculty member's first appointment ∗ •

Remarks

Faculty: a class derived from Entry for a School Directory ∗ faculty entry ∗ inherits a first name, last name, and email address from Entry ∗ additional fields are the faculty member's office, extension, ∗ department, and year of initial aappointment to the school ∗

-

Inherited capabilities include: ∗ Base constructors ∗ Comparison operations depend upon last names, then first names ∗ Formatted orubt abd output ∗

-

Overwritten capabilities include: ∗ Multi-parameter constructor ∗ Formatted print method ∗

-

: files include header (Faculty.h), Implementation (Faculty.cpp)∗

-

Uncomment a main program for unit testing ∗

-

Author

Henry M. Walker ∗

Date

January 11, 2023 ∗

-

Remarks

References ∗

A School Directory as an Example of Object-Oriented Design ∗ `http://localhost/courses/cpp-style-guide/c` `php` ∗

-

Full-parameter constructor ∗

-

Parameters

first	a faculty member's first name ∗
last	a faculty member's last name ∗
eAddress	a faculty member's email address ∗
room	a faculty member's office ∗
ext	the telelphone number extension for the office ∗
department	the faculty member's [primary] department ∗
yr	the year of the faculty member's first appointment ∗ •

4.2.3 Member Function Documentation

4.2.3.1 print()

```
std::ostream & Faculty::print (
             std::ostream & os ) const  [virtual]
```

Remarks

output a format Faculty object ∗

-

Parameters

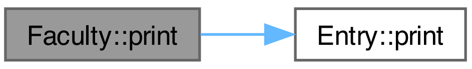
os	output stream which will receive the formatted Faculty data ∗ • @ewmrk by being a virtual function, implementations in subclasses ∗ will be interpreted via polymorphism ∗ •

Returns

formatted string on the given output stream ∗

-

Reimplemented from Entry.
Here is the call graph for this function:



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

- Faculty.h
- Faculty.cpp

4.3 SchoolDirectory Class Reference

Public Member Functions

- SchoolDirectory ()
- void add (Entry person)
- void print ()
- Entry ∗ lookup (std::string first, std::string second)

4.3.1 Detailed Description

Remarks

Example of a School Directory application ∗ Entries take the form of Students, Faculty and Staff ∗

-

Example illustrates a class hierarchy ∗ ∗ Base class: Entry ∗ Subclasses: Student, Faculty, Staff ∗

-

Each class has a header (.h) and implementation (.cpp) files ∗

Other features: overwritten << operator and virtual print ∗ ∗

-

: file: SchoolDirectory.cpp ∗

-

Author

Henry M. Walker ∗

Date

January 11, 2023 ∗

-

Remarks

References ∗

A School Directory as an Example of Object-Oriented Design ∗ `http://localhost/courses/cpp-style-guide/c` `php` ∗

-

4.3.2 Constructor & Destructor Documentation

4.3.2.1 SchoolDirectory()

```
SchoolDirectory::SchoolDirectory ( )  [inline]
```

Remarks

Default constructor (with no parameters) ∗

4.3.3 Member Function Documentation

4.3.3.1 add()

```
void SchoolDirectory::add (
            Entry person )  [inline]
```

Remarks

insert a person into the SchoolDirectory ∗

eirectory entries are maintained in lastname/firstname order ∗

-

Parameters

person	the entry to be inserted into the underlying directory ∗ •

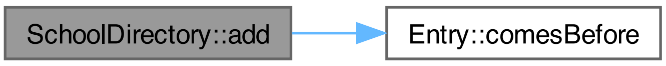
Precondition

entries in the underlying directory are ordered by name ∗

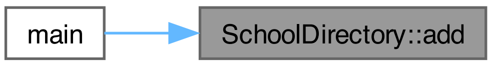
Postcondition

the underlying directory continues to be ordered by name ∗

Here is the call graph for this function:



Here is the caller graph for this function:



4.3.3.2 lookup()

```
Entry ∗ SchoolDirectory::lookup (
            std::string first,
            std::string second )  [inline]
```

Remarks

entries in the directory are searched by first and last name ∗

-

Parameters

first	the first name of a person ∗
last	the last name of a person ∗ •

Precondition

the underlying directory is ordered by last/first name ∗

Returns

if the name is found, a pointer to the entry is returned ∗ if the name is not found, NULL is returned ∗

-

Remarks

searching is performed via a binarysearch ∗

-

Here is the caller graph for this function:

main
SchoolDirectory::lookup

4.3.3.3 print()

`void SchoolDirectory::print ( ) [inline]`

Remarks

entries in the underlying directory are printed to cout ∗ with beginning and end markers ∗

-

Here is the caller graph for this function:

main
SchoolDirectory::print

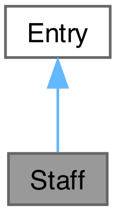
The documentation for this class was generated from the following file:

- SchoolDirectory.cpp

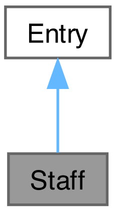
4.4 Staff Class Reference

`#include <Staff.h>`

Inheritance diagram for Staff:



Collaboration diagram for Staff:



Public Member Functions

- Staff (std::string first, std::string last, std::string addr, std::string room, int ext, std::string ttl)
- std::ostream & print (std::ostream &os) const

Public Member Functions inherited from Entry

- Entry ()
- Entry (std::string first, std::string last, std::string eAddress)
- bool equals (std::string first, std::string second)
- bool equals (Entry otherEntry)
- bool comesBefore (std::string first, std::string second)
- bool comesBefore (Entry otherEntry)
- virtual std::ostream & print (std::ostream &os) const
- virtual ∼Entry ()

4.4.1 Detailed Description

Remarks

Student: a class derived from Entry for a School Directory ∗ a staff member's entry ∗ inherits a first name, last name, and email address from Entry ∗ additional fields are the member's office, extension, title ∗

-

Inherited capabilities include: ∗ Base constructors ∗ Comparison operations depend upon last names, then first names ∗ Formatted orubt abd output ∗

-

Overwritten capabilities include: ∗ Multi-parameter constructor ∗ Formatted print method ∗

-

: files include header (Staff.h), Implementation (Staff.cpp) ∗

-

Uncomment a main program for unit testing ∗

-

Author

Henry M. Walker ∗

Date

January 11, 2023 ∗

-

Remarks

References ∗

A School Directory as an Example of Object-Oriented Design ∗ `http://localhost/courses/cpp-style-guide/c` `php` ∗

-

Full-parameter constructor ∗

-

Parameters

first	a staff member's first name ∗
last	a staff member's last name ∗
eAddress	a staff member's email address ∗
room	a staff member's office ∗
ext	the telelphone number extension for the office ∗
title	the staff member's title ∗ •

4.4.2 Constructor & Destructor Documentation

4.4.2.1 Staff()

```
Staff::Staff (
            std::string first,
            std::string last,
            std::string addr,
            std::string room,
            int ext,
            std::string ttl )
```

Remarks

Student: a class derived from Entry for a School Directory ∗ a staff member's entry ∗ inherits a first name, last name, and email address from Entry ∗ additional fields are the member's office, extension, title ∗

-

Inherited capabilities include: ∗ Base constructors ∗ Comparison operations depend upon last names, then first names ∗ Formatted orubt abd output ∗

-

Overwritten capabilities include: ∗ Multi-parameter constructor ∗ Formatted print method ∗

-

: files include header (Staff.h), Implementation (Staff.cpp) ∗

-

Uncomment a main program for unit testing ∗

-

Author

Henry M. Walker ∗

Date

January 11, 2023 ∗

-

Remarks

References ∗

A School Directory as an Example of Object-Oriented Design ∗ http://localhost/courses/cpp-style-guide/c
php ∗

-

4.4.3 Member Function Documentation

4.4.3.1 print()

```
std::ostream & Staff::print (
          std::ostream & os ) const  [virtual]
```

Remarks

output a format Faculty object ∗

-

Parameters

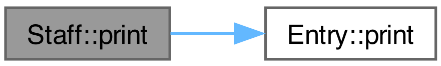
os	output stream which will receive the formatted Faculty data ∗ • @ewmrk by being a virtual function, implementations in subclasses ∗ will be interpreted via polymorphism ∗ •

Returns

formatted string on the given output stream ∗

•

Reimplemented from Entry.
Here is the call graph for this function:

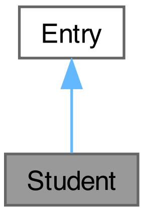


The documentation for this class was generated from the following files:

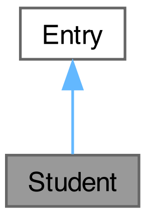
- Staff.h
- Staff.cpp

4.5 Student Class Reference

`#include <Student.h>`
Inheritance diagram for Student:



Collaboration diagram for Student:



Public Member Functions

- Student (std::string first, std::string last, std::string addr, int yr, std::string box)
- std::ostream & print (std::ostream &os) const

Public Member Functions inherited from Entry

- Entry ()
- Entry (std::string first, std::string last, std::string eAddress)
- bool equals (std::string first, std::string second)
- bool equals (Entry otherEntry)
- bool comesBefore (std::string first, std::string second)
- bool comesBefore (Entry otherEntry)
- virtual std::ostream & print (std::ostream &os) const
- virtual ∼Entry ()

4.5.1 Detailed Description

Remarks

Student: a class derived from Entry for a School Directory ∗ a student entry ∗ inherits a first name, last name, and email address from Entry ∗ additional fields are the studentr's year and PO Box ∗

-

Inherited capabilities include: ∗ Base constructors ∗ Comparison operations depend upon last names, then first names ∗ Formatted orubt abd output ∗

-

Overwritten capabilities include: ∗ Multi-parameter constructor ∗ Formatted print method ∗

-

: files include header (Student.h), Implementation (Student.cpp)∗

-

Uncomment a main program for unit testing ∗

-

Author

Henry M. Walker ∗

Date

January 11, 2023 ∗

-

Remarks

References ∗

A School Directory as an Example of Object-Oriented Design ∗ http://localhost/courses/cpp-style-guide/c
php ∗

-

4.5.2 Constructor & Destructor Documentation

4.5.2.1 Student()

```
Student::Student (
            std::string first,
            std::string last,
            std::string addr,
            int yr,
            std::string box )
```

Remarks

Full-parameter constructor ∗

-

Parameters

first	a student's first name ∗
last	a student's last name ∗
eAddress	a student's email address ∗
year	the student's class or expected-graduation year ∗
box	the student's campus post office box ∗ •

4.5.3 Member Function Documentation

4.5.3.1 print()

```
std::ostream & Student::print (
            std::ostream & os ) const  [virtual]
```

Remarks

output a format Faculty object ∗

-

Parameters

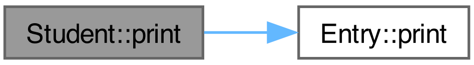
os	output stream which will receive the formatted Faculty data ∗ • @ewmrk by being a virtual function, implementations in subclasses ∗ will be interpreted via polymorphism ∗ •

Returns

formatted string on the given output stream ∗

-

Reimplemented from Entry.
Here is the call graph for this function:



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

- Student.h
- Student.cpp

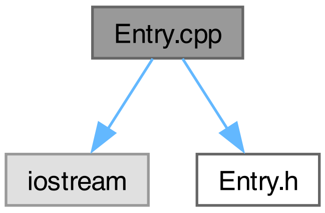
Chapter 5

File Documentation

5.1 Entry.cpp File Reference

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

Include dependency graph for Entry.cpp:



Functions

- std::ostream & operator<< (std::ostream &os, const Entry &ent)
- int main ()

5.1.1 Function Documentation

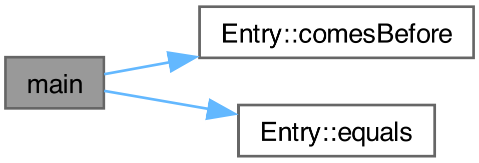
5.1.1.1 main()

```
int main ( )
```

Remarks

main procedure to control processing r uncomment this procedure for unit testing ∗

Here is the call graph for this function:



5.1.1.2 operator<<()

```
std::ostream & operator<< (
            std::ostream & os,
            const Entry & ent )
```

Remarks

overload << for printing an Entry ∗

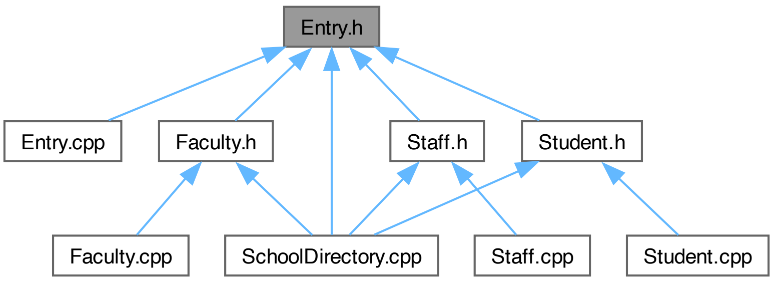
•

use of the virtual print method allows tailored output ∗ by subclasses ∗

•

5.2 Entry.h File Reference

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



Classes

- class Entry

5.3 Entry.h

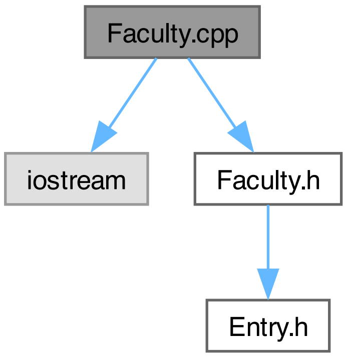
Go to the documentation of this file.

```
#ifndef ENTRY_H
#define ENTRY_H

class Entry {

public:
  // constructors
  Entry () ;

  Entry (std::string first, std::string last, std::string eAddress) ;

  bool equals (std::string first, std::string second) ;

  bool equals (Entry otherEntry) ;

  bool comesBefore (std::string first, std::string second) ;

  bool comesBefore (Entry otherEntry) ;

  virtual
    std::ostream& print (std::ostream &os) const;

  friend std::ostream& operator« (std::ostream &os, const Entry& ent) ;

  virtual
    ~Entry () ;

private:
  // could use "protected" here, so variables may be accessed in subclasses
  std::string firstName;
  std::string lastName;
  std::string eMail;

};

#endif
```

5.4 Faculty.cpp File Reference

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

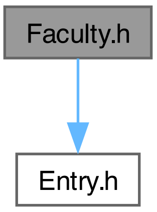
Include dependency graph for Faculty.cpp:



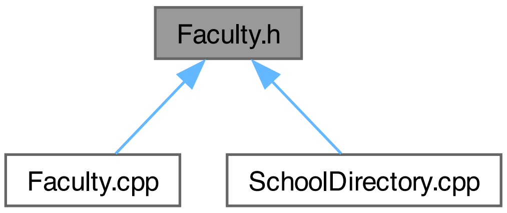
5.5 Faculty.h File Reference

`#include "Entry.h"`

Include dependency graph for Faculty.h:



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



Classes

- class Faculty

5.6 Faculty.h

Go to the documentation of this file.

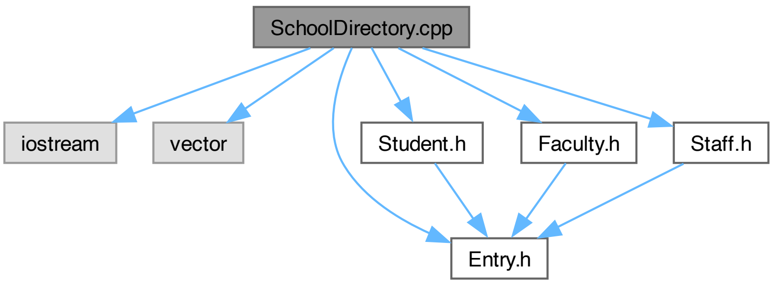
```
#ifndef FACULTY_H
#define FACULTY_H

#include "Entry.h"

// Directory entries specific to faculty
class Faculty : public Entry {
    // Faculty have four special fields

 public:

  Faculty (std::string first, std::string last, std::string addr, std::string room,
           int ext, std::string department, int yr);

  virtual std::ostream& print (std::ostream &os) const ;

      // Faculty have four special fields
private:
  std::string office ;
  int extension ;
```

```
    std::string dept ;
    int firstYear ;

};

#endif
```

5.7 SchoolDirectory.cpp File Reference

```
#include <iostream>
#include <vector>
#include "Entry.h"
#include "Student.h"
#include "Faculty.h"
#include "Staff.h"
```

Include dependency graph for SchoolDirectory.cpp:



Classes

- class SchoolDirectory

Functions

- int main ()

5.7.1 Function Documentation

5.7.1.1 main()

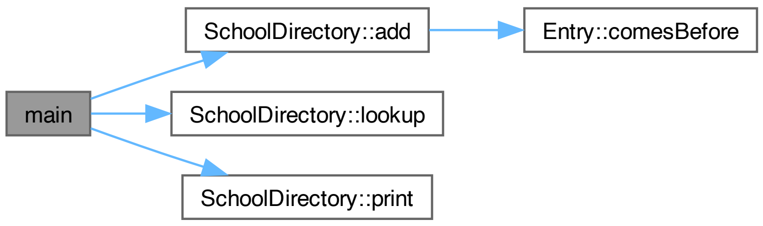
`int main ( )`

Remarks

main performs a reasonable level of testing ∗

-

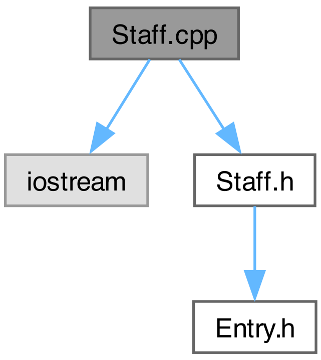
Here is the call graph for this function:



5.8 Staff.cpp File Reference

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

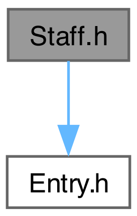
Include dependency graph for Staff.cpp:



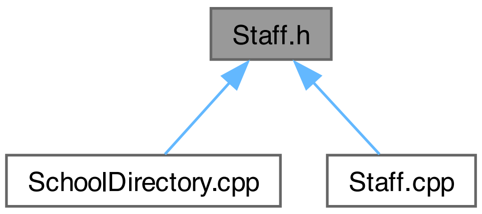
5.9 Staff.h File Reference

```
#include "Entry.h"
```

Include dependency graph for Staff.h:



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



Classes

- class Staff

5.10 Staff.h

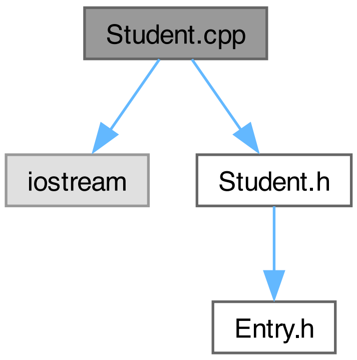
Go to the documentation of this file.

```
#ifndef STAFF_H
#define STAFF_H

#include "Entry.h"

class Staff : public Entry {
 public:
  Staff (std::string first, std::string last, std::string addr,
         std::string room, int ext, std::string ttl) ;

  std::ostream& print (std::ostream &os) const ;

// Staff have two special fields
 private:
  std::string office ;
  int extension ;
  std::string title ;

};

#endif
```

5.11 Student.cpp File Reference

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

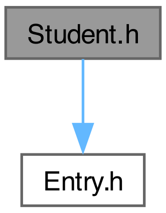
Include dependency graph for Student.cpp:



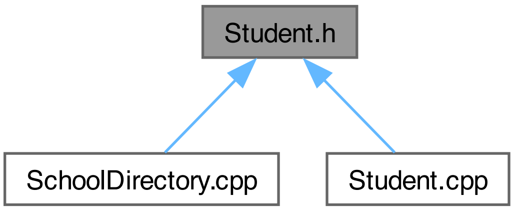
5.12 Student.h File Reference

```
#include "Entry.h"
```

Include dependency graph for Student.h:



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



Classes

- class Student

5.13 Student.h

Go to the documentation of this file.

```
#ifndef STUDENT_H
#define STUDENT_H

#include "Entry.h"


class Student : public Entry {
 public:
  Student (std::string first, std::string last, std::string addr, int yr, std::string box) ;

  std::ostream& print (std::ostream &os) const ;

// Students have two special fields
 private:
    int year;
    std::string POBox;

};

#endif
```

Index