Zip Code Project Pt. 2

Generated by Doxygen 1.12.0

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

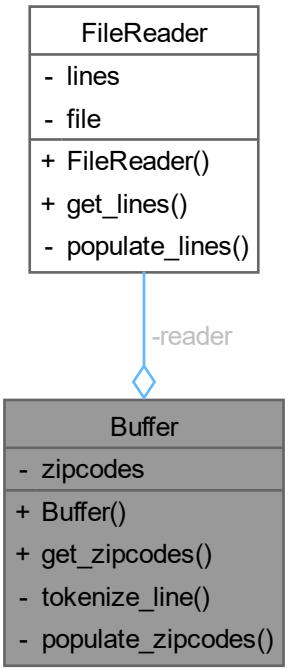
Class Documentation

3.1 Buffer Class Reference

The Buffer class reads zip code data from a file and processes it into a usable format.

`#include <buffer (2).h>`

Collaboration diagram for Buffer:



Public Member Functions

- Buffer (const std::string &)

  Constructs a Buffer object that reads data from the specified file.
- std::vector< ZipCodeData > get_zipcodes ()

  Returns the vector containing all zip code data.

Private Member Functions

- std::tuple< std::string, std::string, std::string, std::string, float, float > tokenize_line (const std::string &)

  Tokenizes a line of CSV data into individual zip code components.
- void populate_zipcodes ()

  Populates the zipcodes vector with data parsed from the file.

Private Attributes

- std::vector< ZipCodeData > zipcodes

  Vector to store all zip code data.
- FileReader reader

  FileReader object to handle file operations.

3.1.1 Detailed Description

The Buffer class reads zip code data from a file and processes it into a usable format.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Buffer()

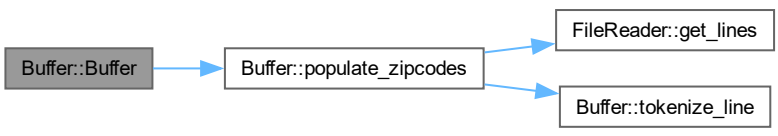
```
Buffer::Buffer (
            const std::string & file)
```

Constructs a Buffer object that reads data from the specified file.

Parameters

file	The path to the input CSV file.

Here is the call graph for this function:



3.1.3 Member Function Documentation

3.1.3.1 get_zipcodes()

```
std::vector< ZipCodeData > Buffer::get_zipcodes ()
```

Returns the vector containing all zip code data.

Returns

A vector of ZipCodeData objects.

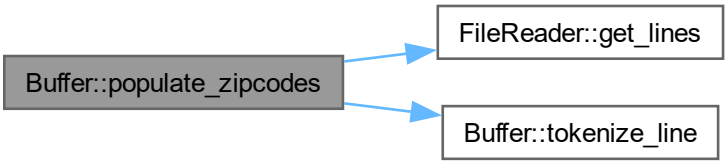
Here is the caller graph for this function:

3.1.3.2 populate_zipcodes()

```
void Buffer::populate_zipcodes ()  [private]
```

Populates the zipcodes vector with data parsed from the file.

Here is the call graph for this function:



Here is the caller graph for this function:

3.1.3.3 tokenize_line()

```
std::tuple< std::string, std::string, std::string, std::string, float, float > Buffer::tokenize↩
_line (
            const std::string & line)  [private]
```

Tokenizes a line of CSV data into individual zip code components.

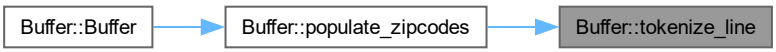
Parameters

line	The input string representing a single line of CSV data.

Returns

A tuple containing the zip code, place name, state, county, latitude, and longitude.

Here is the caller graph for this function:



3.1.4 Member Data Documentation

3.1.4.1 reader

```
FileReader Buffer::reader  [private]
```

FileReader object to handle file operations.

3.1.4.2 zipcodes

```
std::vector<ZipCodeData> Buffer::zipcodes  [private]
```

Vector to store all zip code data.

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

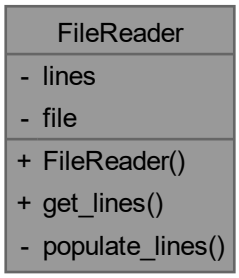
- buffer (2).h
- buffer (2).cpp

3.2 FileReader Class Reference

The FileReader class handles reading lines from a file.

`#include <filereader (2).h>`

Collaboration diagram for FileReader:



Public Member Functions

- FileReader (const std::string &)

  Constructs a FileReader object and opens the specified file.
- std::vector< std::string > get_lines ()

  Returns the lines read from the file.

Private Member Functions

- void populate_lines ()

  Populates the lines vector by reading each line from the file.

Private Attributes

- std::vector< std::string > lines

  Vector to store lines read from the file.
- std::ifstream file

  Input file stream.

Friends

- std::ostream & operator<< (std::ostream &outputstream, const FileReader &reader)

  Overloads the << operator to print all lines of the file to an output stream.

3.2.1 Detailed Description

The FileReader class handles reading lines from a file.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 FileReader()

```
FileReader::FileReader (
            const std::string & input_file)
```

Constructs a FileReader object and opens the specified file.

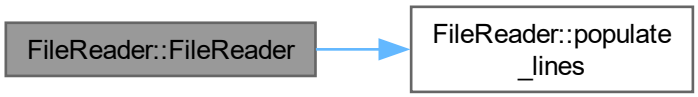
Parameters

input_file	The path to the input file.

Exceptions

std::runtime_error	If the file cannot be opened.

Here is the call graph for this function:



3.2.3 Member Function Documentation

3.2.3.1 get_lines()

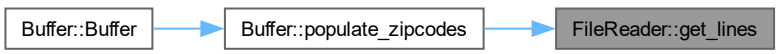
```
std::vector< std::string > FileReader::get_lines ()
```

Returns the lines read from the file.

Returns

A vector of strings representing each line in the file.

Here is the caller graph for this function:

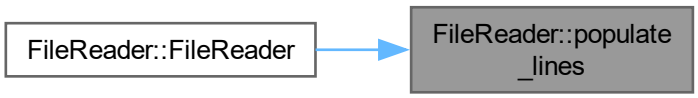


3.2.3.2 populate_lines()

```
void FileReader::populate_lines ()  [private]
```

Populates the lines vector by reading each line from the file.

Here is the caller graph for this function:



3.2.4 Friends And Related Symbol Documentation

3.2.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & outputstream,
            const FileReader & reader)  [friend]
```

Overloads the << operator to print all lines of the file to an output stream.

Parameters

outputstream	The output stream.
reader	The FileReader object containing the lines to print.

Returns

The output stream after the lines are written.

3.2.5 Member Data Documentation

3.2.5.1 file

```
std::ifstream FileReader::file  [private]
```

Input file stream.

3.2.5.2 lines

```
std::vector<std::string> FileReader::lines  [private]
```

Vector to store lines read from the file.

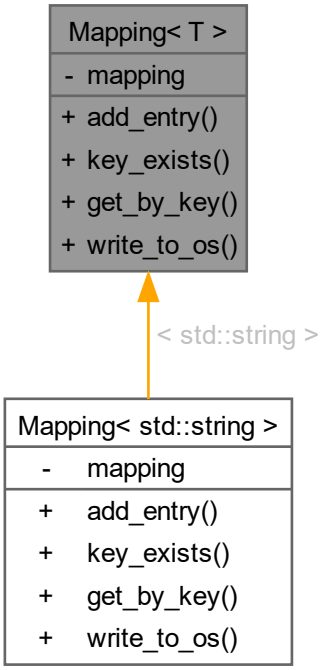
The documentation for this class was generated from the following files:

- filereader (2).h
- filereader (2).cpp

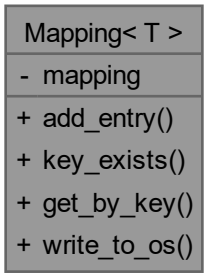
3.3 Mapping< T > Class Template Reference

```
#include <mapping.h>
```

Inheritance diagram for Mapping< T >:



Collaboration diagram for Mapping< T >:



Public Member Functions

- void add_entry (const T &, const ZipCodeData &)
- bool key_exists (const T &)
- std::vector< ZipCodeData > get_by_key (const T &)
- void write_to_os (std::ostream &, const T &)

Private Attributes

- std::map< T, std::vector< ZipCodeData > > mapping

3.3.1 Member Function Documentation

3.3.1.1 add_entry()

```
template<typename T >
void Mapping< T >::add_entry (
            const T & ,
            const ZipCodeData & )
```

Here is the caller graph for this function:

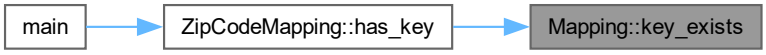
3.3.1.2 get_by_key()

```
template<typename T >
std::vector< ZipCodeData > Mapping< T >::get_by_key (
            const T & )
```

3.3.1.3 key_exists()

```
template<typename T >
bool Mapping< T >::key_exists (
            const T & )
```

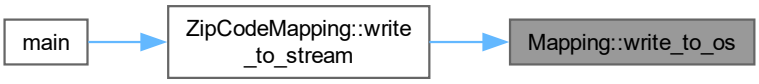
Here is the caller graph for this function:



3.3.1.4 write_to_os()

```
template<typename T >
void Mapping< T >::write_to_os (
            std::ostream & ,
            const T & )
```

Here is the caller graph for this function:



3.3.2 Member Data Documentation

3.3.2.1 mapping

```
template<typename T >
std::map<T, std::vector<ZipCodeData> > Mapping< T >::mapping  [private]
```

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

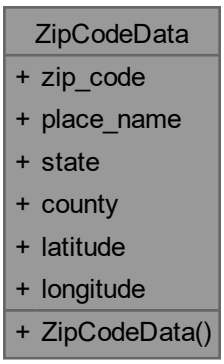
- mapping.h

3.4 ZipCodeData Struct Reference

The ZipCodeData struct holds data for a single zip code, including its coordinates and place information.

`#include <zipcode (2).h>`

Collaboration diagram for ZipCodeData:



Public Member Functions

- ZipCodeData (std::tuple< std::string, std::string, std::string, std::string, float, float >)

  Constructs a ZipCodeData object from a tuple containing zip code details.

Public Attributes

- std::string zip_code

  The zip code.
- std::string place_name

  The name of the place corresponding to the zip code.
- std::string state

  The state (two-character abbreviation).
- std::string county

  The county of the place.
- float latitude

  The latitude coordinate.
- float longitude

  The longitude coordinate.

Friends

- std::ostream & operator<< (std::ostream &outputstream, const ZipCodeData &zipcode)

  Overloads the << operator to print the ZipCodeData to an output stream.

3.4.1 Detailed Description

The ZipCodeData struct holds data for a single zip code, including its coordinates and place information.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 ZipCodeData()

```
ZipCodeData::ZipCodeData (
            std::tuple< std::string, std::string, std::string, std::string, float, float >
tuple)
```

Constructs a ZipCodeData object from a tuple containing zip code details.

Parameters

tuple	A tuple containing zip code, place name, state, county, latitude, and longitude.

3.4.3 Friends And Related Symbol Documentation

3.4.3.1 operator<<

```
std::ostream & operator<< (
            std::ostream & outputstream,
            const ZipCodeData & zipcode)  [friend]
```

Overloads the << operator to print the ZipCodeData to an output stream.

Parameters

outputstream	The output stream.
zipcode	The ZipCodeData object to be printed.

Returns

The output stream after the zip code data is written.

3.4.4 Member Data Documentation

3.4.4.1 county

```
std::string ZipCodeData::county
```

The county of the place.

3.4.4.2 latitude

```
float ZipCodeData::latitude
```

The latitude coordinate.

3.4.4.3 longitude

```
float ZipCodeData::longitude
```

The longitude coordinate.

3.4.4.4 place_name

```
std::string ZipCodeData::place_name
```

The name of the place corresponding to the zip code.

3.4.4.5 state

```
std::string ZipCodeData::state
```

The state (two-character abbreviation).

3.4.4.6 zip_code

```
std::string ZipCodeData::zip_code
```

The zip code.

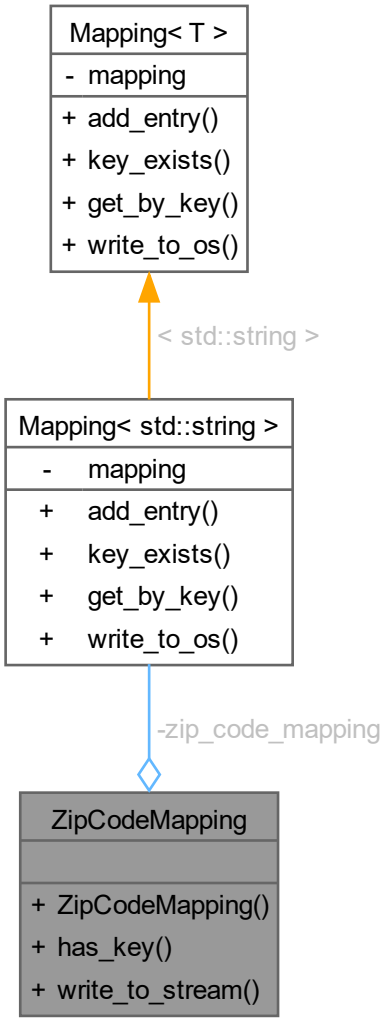
The documentation for this struct was generated from the following files:

- zipcode (2).h
- zipcode (2).cpp

3.5 ZipCodeMapping Class Reference

`#include <zipcodemapping.h>`

Collaboration diagram for ZipCodeMapping:



Public Member Functions

- ZipCodeMapping (const std::vector< ZipCodeData > &)
- bool has_key (const std::string &)
- void write_to_stream (std::ostream &, const std::string &)

Private Attributes

- Mapping< std::string > zip_code_mapping

3.5.1 Constructor & Destructor Documentation

3.5.1.1 ZipCodeMapping()

```
ZipCodeMapping::ZipCodeMapping (
            const std::vector< ZipCodeData > & data)
```

Here is the call graph for this function:

3.5.2 Member Function Documentation

3.5.2.1 has_key()

```
bool ZipCodeMapping::has_key (
            const std::string & zipcode)
```

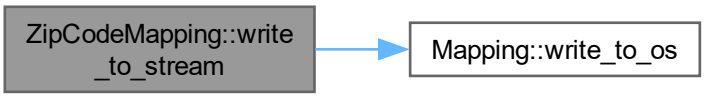
Here is the call graph for this function:

Here is the caller graph for this function:

3.5.2.2 write_to_stream()

```
void ZipCodeMapping::write_to_stream (
            std::ostream & os,
            const std::string & zipcode)
```

Here is the call graph for this function:



Here is the caller graph for this function:

3.5.3 Member Data Documentation

3.5.3.1 zip_code_mapping

```
Mapping<std::string> ZipCodeMapping::zip_code_mapping  [private]
```

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

- zipcodemapping.h
- zipcodemapping.cpp

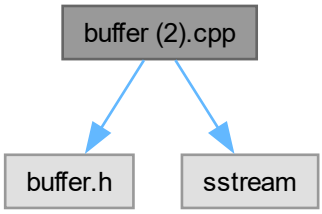
Chapter 4

File Documentation

4.1 buffer (2).cpp File Reference

```
#include "buffer.h"
#include <sstream>
```

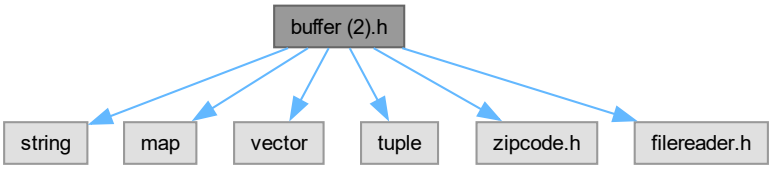
Include dependency graph for buffer (2).cpp:



4.2 buffer (2).h File Reference

```
#include <string>
#include <map>
#include <vector>
#include <tuple>
#include "zipcode.h"
```

`#include "filereader.h"`
Include dependency graph for buffer (2).h:



Classes

- class Buffer

  The Buffer class reads zip code data from a file and processes it into a usable format.

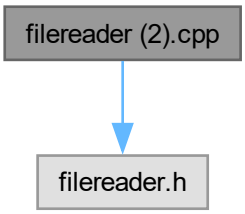
4.3 buffer (2).h

Go to the documentation of this file.

```
00001
00004 #ifndef BUFFER_H
00005 #define BUFFER_H
00006
00007 #include <string>
00008 #include <map>
00009 #include <vector>
00010 #include <tuple>
00011
00012 #include "zipcode.h"
00013 #include "filereader.h"
00014
00018 class Buffer {
00019     std::vector<ZipCodeData> zipcodes;
00020     FileReader reader;
00021
00028     std::tuple<std::string, std::string, std::string, std::string, float, float> tokenize_line(const
      std::string&);
00029
00033     void populate_zipcodes();
00034
00035 public:
00041     Buffer(const std::string&);
00042
00048     std::vector<ZipCodeData> get_zipcodes();
00049 };
00050
00051 #endif // BUFFER_H
```

4.4 filereader (2).cpp File Reference

```
#include "filereader.h"
```
Include dependency graph for filereader (2).cpp:



Functions

- std::ostream & operator<< (std::ostream &outputstream, const FileReader &reader)

4.4.1 Function Documentation

4.4.1.1 operator<<()

```
std::ostream & operator<< (
            std::ostream & outputstream,
            const FileReader & reader)
```

Parameters

outputstream	The output stream.
reader	The FileReader object containing the lines to print.

Returns

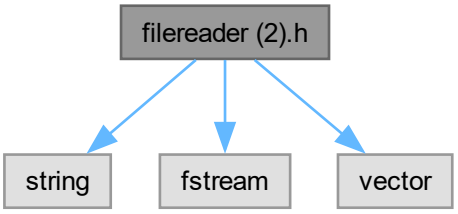
The output stream after the lines are written.

4.5 filereader (2).h File Reference

```
#include <string>
#include <fstream>
```

```
#include <vector>
```

Include dependency graph for filereader (2).h:



Classes

- class FileReader

  The FileReader class handles reading lines from a file.

4.6 filereader (2).h

Go to the documentation of this file.

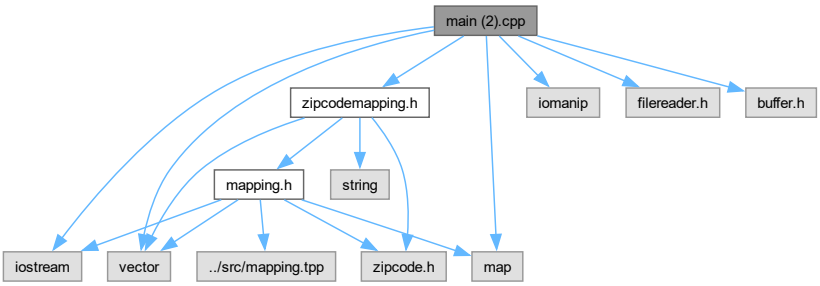
```
00001
00004 #ifndef FILE_READER_H
00005 #define FILE_READER_H
00006
00007 #include <string>
00008 #include <fstream>
00009 #include <vector>
00010
00014 class FileReader
00015 {
00016     std::vector<std::string> lines;
00017     std::ifstream file;
00018
00022     void populate_lines();
00023
00024 public:
00031     FileReader(const std::string&);
00032
00038     std::vector<std::string> get_lines();
00039
00047     friend std::ostream& operator<<(std::ostream&, const FileReader&);
00048 };
00049
00050 #endif // FILE_READER_H
```

4.7 main (2).cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <vector>
#include <map>
#include "zipcodemapping.h"
#include "filereader.h"
```

```
#include "buffer.h"
```

Include dependency graph for main (2).cpp:



Functions

- int main (int argc, char ∗argv[ ])

  Main function that reads zip code data from a CSV file, parses it, and prints the extreme zip codes (East, West, North, South) for each state.

4.7.1 Function Documentation

4.7.1.1 main()

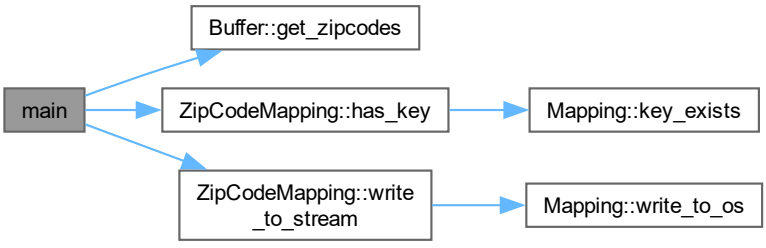
```
int main (
            int argc,
            char * argv[ ])
```

Main function that reads zip code data from a CSV file, parses it, and prints the extreme zip codes (East, West, North, South) for each state.

Returns

int Returns 0 if the program executes successfully.

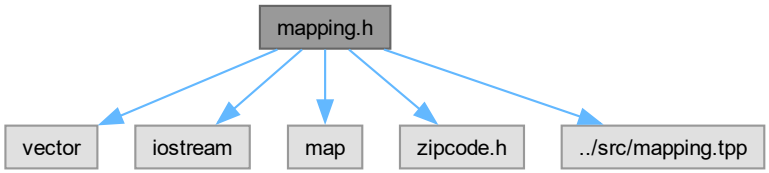
Here is the call graph for this function:



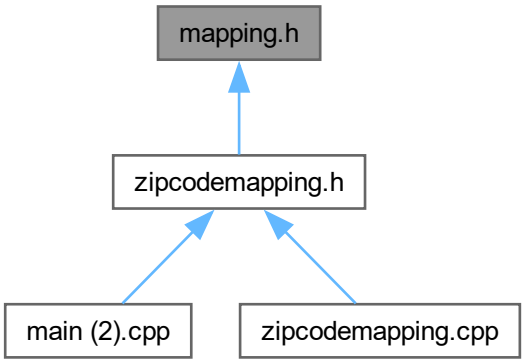
4.8 mapping.h File Reference

```
#include <vector>
#include <iostream>
#include <map>
#include "zipcode.h"
#include "../src/mapping.tpp"
```

Include dependency graph for mapping.h:



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



Classes

- class Mapping< T >

4.9 mapping.h

Go to the documentation of this file.

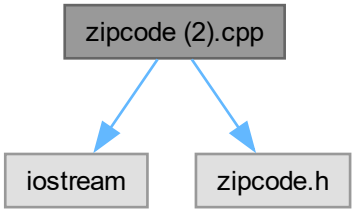
```
00001 #ifndef MAPPING_H
00002 #define MAPPING_H
00003
00004 #include <vector>
```

```
00005 #include <iostream>
00006 #include <map>
00007 #include "zipcode.h"
00008
00009 template <typename T>
00010 class Mapping {
00011     std::map<T, std::vector<ZipCodeData» mapping;
00012 public:
00013     void add_entry(const T&, const ZipCodeData&);
00014     bool key_exists(const T&);
00015     std::vector<ZipCodeData> get_by_key(const T&);
00016     void write_to_os(std::ostream&, const T&);
00017
00018 };
00019
00020 #include "../src/mapping.tpp"
00021
00022 #endif // MAPPING_H
```

4.10 zipcode (2).cpp File Reference

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

Include dependency graph for zipcode (2).cpp:



Functions

- std::ostream & operator<< (std::ostream &outputstream, const ZipCodeData &zipcode)

4.10.1 Function Documentation

4.10.1.1 operator<<()

```
std::ostream & operator<< (
            std::ostream & outputstream,
            const ZipCodeData & zipcode)
```

Parameters

outputstream	The output stream.
zipcode	The ZipCodeData object to be printed.

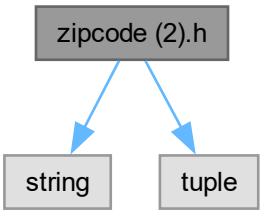
Returns

The output stream after the zip code data is written.

4.11 zipcode (2).h File Reference

```
#include <string>
#include <tuple>
```

Include dependency graph for zipcode (2).h:



Classes

- struct ZipCodeData

  The ZipCodeData struct holds data for a single zip code, including its coordinates and place information.

4.12 zipcode (2).h

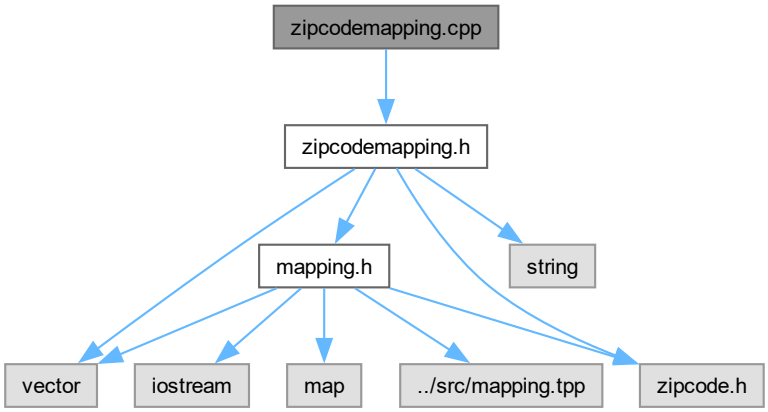
Go to the documentation of this file.

```
00001
00004 #ifndef ZIP_CODE_H
00005 #define ZIP_CODE_H
00006
00007 #include <string>
00008 #include <tuple>
00009
00013 struct ZipCodeData
00014 {
00015     std::string zip_code;
00016     std::string place_name;
00017     std::string state;
00018     std::string county;
00019     float latitude;
00020     float longitude;
00021
00027     ZipCodeData(std::tuple<std::string, std::string, std::string, std::string, float, float>);
00028
00036     friend std::ostream& operator<<(std::ostream&, const ZipCodeData&);
00037 };
00038
00039 #endif // ZIP_CODE_H
```

4.13 zipcodemapping.cpp File Reference

```
#include "zipcodemapping.h"
```

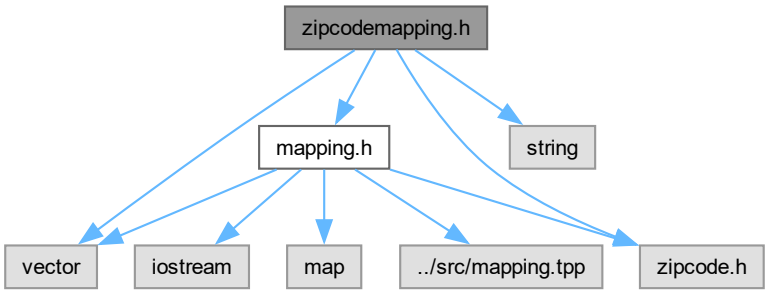
Include dependency graph for zipcodemapping.cpp:



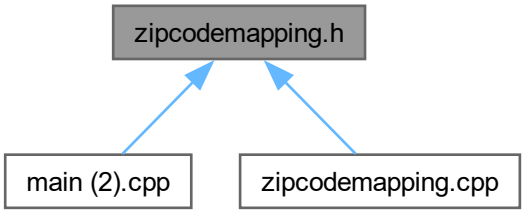
4.14 zipcodemapping.h File Reference

```
#include "mapping.h"
#include "zipcode.h"
#include <string>
#include <vector>
```

Include dependency graph for zipcodemapping.h:



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



Classes

- class ZipCodeMapping

4.15 zipcodemapping.h

Go to the documentation of this file.

```
00001 #ifndef ZIP_CODE_MAPPING_H
00002 #define ZIP_CODE_MAPPING_H
00003
00004 #include "mapping.h"
00005 #include "zipcode.h"
00006 #include <string>
00007 #include <vector>
00008
00009 class ZipCodeMapping {
00010     Mapping<std::string> zip_code_mapping;
00011 public:
00012     ZipCodeMapping(const std::vector<ZipCodeData>&);
00013     bool has_key(const std::string&);
00014     void write_to_stream(std::ostream&, const std::string&);
00015 };
00016
00017 #endif // ZIP_CODE_MAPPING_H
```