

My House Market Analysis

0.2

Generated by Doxygen 1.8.13

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	HousePrice Class Reference	5
3.1.1	Constructor & Destructor Documentation	6
3.1.1.1	HousePrice() [1/2]	6
3.1.1.2	HousePrice() [2/2]	6
3.1.2	Member Function Documentation	6
3.1.2.1	getPrice()	7
3.1.2.2	getState()	7
3.1.3	Friends And Related Function Documentation	7
3.1.3.1	operator<<	7
3.1.4	Member Data Documentation	7
3.1.4.1	city	8
3.1.4.2	id	8
3.1.4.3	number	8
3.1.4.4	postalCode	8
3.1.4.5	price	8
3.1.4.6	state	8
3.1.4.7	street	8

4 File Documentation	9
4.1 HouseFunc.cpp File Reference	9
4.1.1 Function Documentation	10
4.1.1.1 houseMarketPerState()	10
4.1.1.2 houseMarketValues()	10
4.1.1.3 readCSV()	11
4.2 HouseFunc.h File Reference	11
4.2.1 Function Documentation	12
4.2.1.1 houseMarketInfo()	12
4.2.1.2 readCSV()	12
4.3 HousePrice.cpp File Reference	13
4.3.1 Function Documentation	13
4.3.1.1 operator<<()	13
4.4 HousePrice.h File Reference	14
4.5 main.cpp File Reference	14
4.5.1 Function Documentation	15
4.5.1.1 main()	15
Index	17

Chapter 1

Class Index

1.1 Class List

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

HousePrice	5
--------------------------------------	---

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

HouseFunc.cpp	9
HouseFunc.h	11
HousePrice.cpp	13
HousePrice.h	14
main.cpp	14

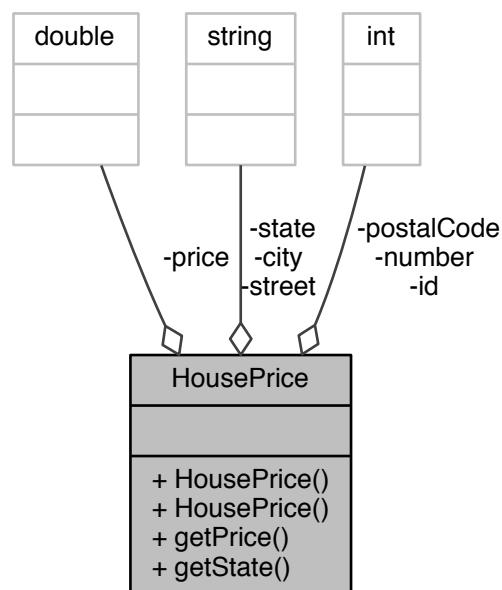
Chapter 3

Class Documentation

3.1 HousePrice Class Reference

```
#include <HousePrice.h>
```

Collaboration diagram for HousePrice:



Public Member Functions

- `HousePrice ()`
- `HousePrice (int id, int number, const string &street, const string &city, const string &state, int postalCode, double price)`
- `double getPrice () const`
- `const string &getState () const`

Private Attributes

- int `id`
- int `number`
- string `street`
- string `city`
- string `state`
- int `postalCode`
- double `price`

Friends

- ostream & `operator<<` (ostream &os, const `HousePrice` &price)

3.1.1 Constructor & Destructor Documentation

3.1.1.1 `HousePrice()` [1/2]

```
HousePrice::HousePrice ( )
```

3.1.1.2 `HousePrice()` [2/2]

```
HousePrice::HousePrice (
    int id,
    int number,
    const string & street,
    const string & city,
    const string & state,
    int postalCode,
    double price )
```

3.1.2 Member Function Documentation

3.1.2.1 getPrice()

```
double HousePrice::getPrice ( ) const
```

Here is the caller graph for this function:



3.1.2.2 getState()

```
const string & HousePrice::getState ( ) const
```

Here is the caller graph for this function:



3.1.3 Friends And Related Function Documentation

3.1.3.1 operator<<

```
ostream& operator<< (  
    ostream & os,  
    const HousePrice & price ) [friend]
```

3.1.4 Member Data Documentation

3.1.4.1 city

```
string HousePrice::city [private]
```

3.1.4.2 id

```
int HousePrice::id [private]
```

3.1.4.3 number

```
int HousePrice::number [private]
```

3.1.4.4 postalCode

```
int HousePrice::postalCode [private]
```

3.1.4.5 price

```
double HousePrice::price [private]
```

3.1.4.6 state

```
string HousePrice::state [private]
```

3.1.4.7 street

```
string HousePrice::street [private]
```

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

- [HousePrice.h](#)
- [HousePrice.cpp](#)

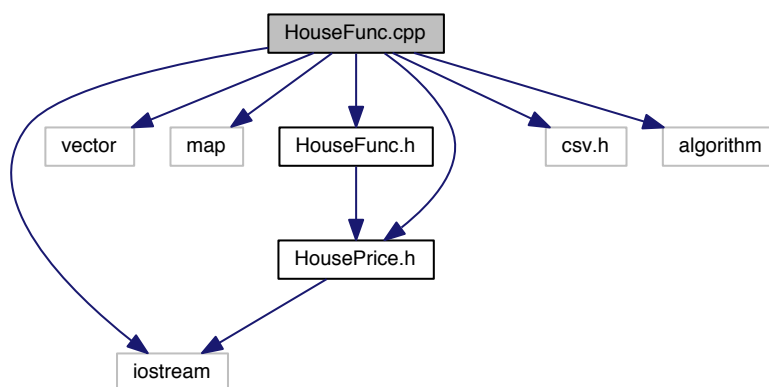
Chapter 4

File Documentation

4.1 HouseFunc.cpp File Reference

```
#include <iostream>
#include <vector>
#include <map>
#include "HouseFunc.h"
#include "HousePrice.h"
#include "csv.h"
#include <algorithm>
```

Include dependency graph for HouseFunc.cpp:



Functions

- void [readCSV](#) (const string fileIn, vector< [HousePrice](#) > &hp)
- void [houseMarketValues](#) (vector< [HousePrice](#) > &hp)
- void [houseMarketPerState](#) (vector< [HousePrice](#) > &hp)

4.1.1 Function Documentation

4.1.1.1 houseMarketPerState()

```
void houseMarketPerState (
    vector< HousePrice > & hp )
```

Sort the Vector of [HousePrice](#) Objects by state. Display the number of houses per state. It uses a map to count instances per state.

Parameters

<i>hp</i>	Reference to vector<HousePrice>
-----------	---------------------------------

Here is the call graph for this function:



4.1.1.2 houseMarketValues()

```
void houseMarketValues (
    vector< HousePrice > & hp )
```

Sort the Vector of [HousePrice](#) Objects by price. Display the most affordable and the most expensive house from the vector.

Parameters

<i>hp</i>	Reference to vector<HousePrice>
-----------	---------------------------------

Here is the call graph for this function:



4.1.1.3 readCSV()

```

void readCSV (
    const string fileIn,
    vector< HousePrice > & hp )
  
```

Read CSV Mock Data. Parses each record in the file to an object of [HousePrice](#) house. Then, populate a vector of [HousePrice](#) objects with all the records. It uses special library to parse CSV file "csv.h"

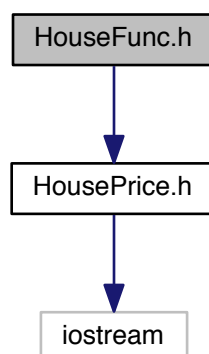
Parameters

<i>fileIn</i>	File name <string>
<i>hp</i>	Reference to Vector of HousePrice Objects

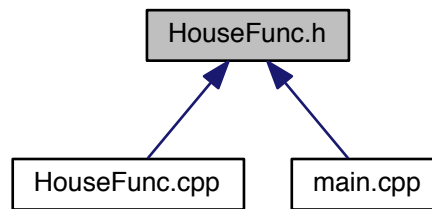
4.2 HouseFunc.h File Reference

```
#include "HousePrice.h"
```

Include dependency graph for HouseFunc.h:



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



Functions

- void [readCSV](#) (const string fileIn, vector< [HousePrice](#) > &hp)
- void [houseMarketInfo](#) (vector< [HousePrice](#) > &hp)

4.2.1 Function Documentation

4.2.1.1 [houseMarketInfo\(\)](#)

```
void houseMarketInfo (
    vector< HousePrice > & hp )
```

4.2.1.2 [readCSV\(\)](#)

```
void readCSV (
    const string fileIn,
    vector< HousePrice > & hp )
```

Read CSV Mock Data. Parses each record in the file to an object of [HousePrice](#) house. Then, populate a vector of [HousePrice](#) objects with all the records. It uses special library to parse CSV file "csv.h"

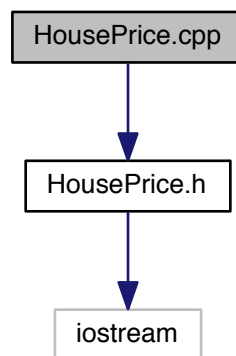
Parameters

<i>fileIn</i>	File name <string>
<i>hp</i>	Reference to Vector of HousePrice Objects

4.3 HousePrice.cpp File Reference

```
#include "HousePrice.h"
```

Include dependency graph for HousePrice.cpp:



Functions

- ostream & `operator<<` (ostream &os, const `HousePrice` &price)

4.3.1 Function Documentation

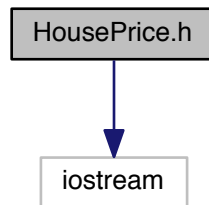
4.3.1.1 `operator<<()`

```
ostream& operator<< (  
    ostream & os,  
    const HousePrice & price )
```

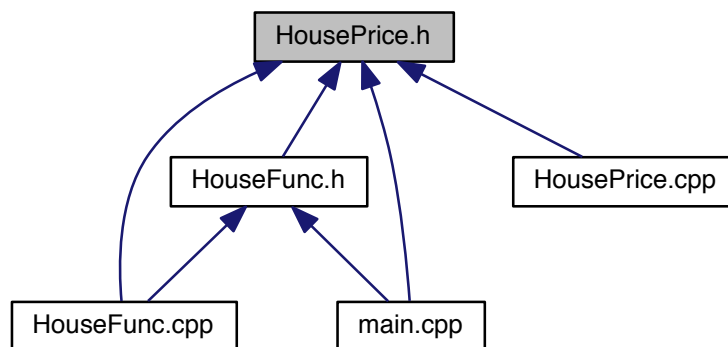
4.4 HousePrice.h File Reference

```
#include <iostream>
```

Include dependency graph for HousePrice.h:



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



Classes

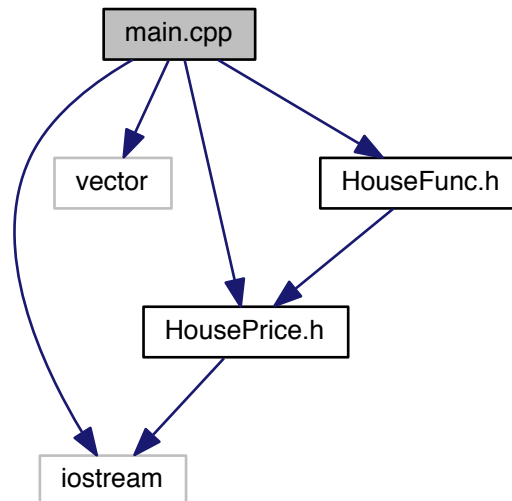
- class [HousePrice](#)

4.5 main.cpp File Reference

```
#include <iostream>
#include <vector>
#include "HousePrice.h"
```

```
#include "HouseFunc.h"
```

Include dependency graph for main.cpp:



Functions

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

4.5.1 Function Documentation

4.5.1.1 main()

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

Main Function

Returns

0 no error, error otherwise

Index

- city
 - HousePrice, 7
- getPrice
 - HousePrice, 6
- getState
 - HousePrice, 7
- HouseFunc.cpp, 9
 - houseMarketPerState, 10
 - houseMarketValues, 10
 - readCSV, 11
- HouseFunc.h, 11
 - houseMarketInfo, 12
 - readCSV, 12
- houseMarketInfo
 - HouseFunc.h, 12
- houseMarketPerState
 - HouseFunc.cpp, 10
- houseMarketValues
 - HouseFunc.cpp, 10
- HousePrice, 5
 - city, 7
 - getPrice, 6
 - getState, 7
 - HousePrice, 6
 - id, 8
 - number, 8
 - operator<<, 7
 - postalCode, 8
 - price, 8
 - state, 8
 - street, 8
- HousePrice.cpp, 13
 - operator<<, 13
- HousePrice.h, 14
- id
 - HousePrice, 8
- main
 - main.cpp, 15
- main.cpp, 14
 - main, 15
- number
 - HousePrice, 8
- operator<<
 - HousePrice, 7
 - HousePrice.cpp, 13
- postalCode
 - HousePrice, 8
- price
 - HousePrice, 8
- readCSV
 - HouseFunc.cpp, 11
 - HouseFunc.h, 12
- state
 - HousePrice, 8
- street
 - HousePrice, 8