Project 1 Zip Code - Team 4

1.0

Generated by Doxygen 1.14.0

Chapter 1

Bug List

File main1.cpp

None that we know of right now.

File main2.cpp

None that we know of right now.

File readCSV.cpp

None that we know of right now.

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:

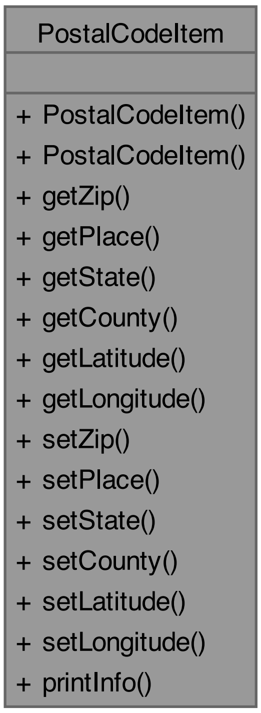
Chapter 4

Class Documentation

4.1 PostalCodeItem Class Reference

```
#include <PostalCodeItem.h>
```

Collaboration diagram for PostalCodeItem:



Public Member Functions

- PostalCodeItem ()

  Default constructor initializing member variables to default values.
- PostalCodeItem (int z, const string &p, const string &s, const string &c, double lat, double lon)

  Parameterized constructor to initialize a PostalCodeItem with specific values.
- int getZip () const

  Get the ZIP code of the postal code item.
- string getPlace () const

  Get the place name of the postal code item.
- string getState () const

  Get the state name of the postal code item.
- string getCounty () const

  Get the county name of the postal code item.
- double getLatitude () const

  Get the latitude of the postal code item.
- double getLongitude () const

  Get the longitude of the postal code item.
- void setZip (int newZip)

  Set the ZIP code of the postal code item.
- void setPlace (const string &newPlace)

  Set the place name of the postal code item.
- void setState (const string &newState)

  Set the state name of the postal code item.
- void setCounty (const string &newCounty)

  Set the county name of the postal code item.
- void setLatitude (double newLat)

  Set the latitude of the postal code item.
- void setLongitude (double newLon)

  Set the longitude of the postal code item.
- void printInfo () const

  Print the postal code item's information in a formatted manner.

4.1.1 Detailed Description

Definition at line 24 of file PostalCodeItem.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 PostalCodeItem() [1/2]

```
PostalCodeItem::PostalCodeItem ()
```

Default constructor initializing member variables to default values.

zip is set to 0, place, state, and county are set to empty strings, and latitude and longitude are set to 0.0. This ensures that a PostalCodeItem object starts with a known state.

Note

This constructor can be used to create an empty PostalCodeItem object, which can later be populated with actual data using the setter methods.

See also

PostalCodeItem(int, const string&, const string&, const string&, double, double)

setZip(int)

setPlace(const string&)

setState(const string&)

setCounty(const string&)

setLatitude(double)

setLongitude(double)

printInfo() const

Definition at line 46 of file PostalCodeItem.cpp.

4.1.2.2 PostalCodeItem() [2/2]

```
PostalCodeItem::PostalCodeItem (
              int z,
              const string & p,
              const string & s,
              const string & c,
              double lat,
              double lon)
```

Parameterized constructor to initialize a PostalCodeItem with specific values.

Parameters

z	The ZIP code (integer).
p	The place name (string).
s	The state name (string).
c	The county name (string).
lat	The latitude (double).
lon	The longitude (double). This constructor allows for the creation of a fully initialized PostalCodeItem object.

Note

Ensure that the provided values are valid and meaningful for the postal code entry.

Definition at line 67 of file PostalCodeItem.cpp.

4.1.3 Member Function Documentation

4.1.3.1 getZip()

```
int PostalCodeItem::getZip () const
```

Get the ZIP code of the postal code item.

Returns

The ZIP code as an integer.

Definition at line 81 of file PostalCodeItem.cpp.

4.1.3.2 getPlace()

```
string PostalCodeItem::getPlace () const
```

Get the place name of the postal code item.

Returns

The place name as a string.

Definition at line 90 of file PostalCodeItem.cpp.

4.1.3.3 getState()

```
string PostalCodeItem::getState () const
```

Get the state name of the postal code item.

Returns

The state name as a string.

Definition at line 99 of file PostalCodeItem.cpp.

4.1.3.4 getCounty()

```
string PostalCodeItem::getCounty () const
```

Get the county name of the postal code item.

Returns

The county name as a string.

Note

County names may vary in format and length depending on the region. Ensure that the county name is correctly formatted for display or processing.

Definition at line 110 of file PostalCodeItem.cpp.

4.1.3.5 getLatitude()

```
double PostalCodeItem::getLatitude () const
```

Get the latitude of the postal code item.

Returns

The latitude as a double.

Note

Latitude values are typically in the range of -90 to 90 degrees.

Definition at line 120 of file PostalCodeItem.cpp.

4.1.3.6 getLongitude()

```
double PostalCodeItem::getLongitude () const
```

Get the longitude of the postal code item.

Returns

The longitude as a double.

Note

Longitude values are typically in the range of -180 to 180 degrees.

Definition at line 130 of file PostalCodeItem.cpp.

4.1.3.7 setZip()

```
void PostalCodeItem::setZip (
             int newZip)
```

Set the ZIP code of the postal code item.

Parameters

newZip	The new ZIP code to be set (integer).

Note

Ensure that the new ZIP code is a valid integer value.

Definition at line 140 of file PostalCodeItem.cpp.

4.1.3.8 setPlace()

```
void PostalCodeItem::setPlace (
             const string & newPlace)
```

Set the place name of the postal code item.

Parameters

newPlace	The new place name to be set (string).

Note

Ensure that the new place name is a valid string value.

Definition at line 150 of file PostalCodeItem.cpp.

4.1.3.9 setState()

```
void PostalCodeItem::setState (
             const string & newState)
```

Set the state name of the postal code item.

Parameters

newState	The new state name to be set (string).

Note

Ensure that the new state name is a valid string value.

Definition at line 160 of file PostalCodeItem.cpp.

4.1.3.10 setCounty()

```
void PostalCodeItem::setCounty (
             const string & newCounty)
```

Set the county name of the postal code item.

Parameters

newCounty	The new county name to be set (string).

Note

Ensure that the new county name is a valid string value.

Definition at line 170 of file PostalCodeItem.cpp.

4.1.3.11 setLatitude()

```
void PostalCodeItem::setLatitude (
             double newLat)
```

Set the latitude of the postal code item.

Parameters

newLat	The new latitude to be set (double).

Note

Ensure that the new latitude is within the valid range of -90 to 90 degrees. Invalid latitude values may lead to incorrect geographical representations.

Definition at line 181 of file PostalCodeItem.cpp.

4.1.3.12 setLongitude()

```
void PostalCodeItem::setLongitude (
             double newLon)
```

Set the longitude of the postal code item.

Parameters

newLon	The new longitude to be set (double).

Note

Ensure that the new longitude is within the valid range of -180 to 180 degrees. Invalid longitude values may lead to incorrect geographical representations.

Definition at line 192 of file PostalCodeItem.cpp.

4.1.3.13 printInfo()

```
void PostalCodeItem::printInfo () const
```

Print the postal code item's information in a formatted manner.

The information includes ZIP code, place name, state, county, latitude, and longitude. The output is aligned in columns for better readability.

Note

This method uses standard output (cout) to display the information.

Definition at line 203 of file PostalCodeItem.cpp.

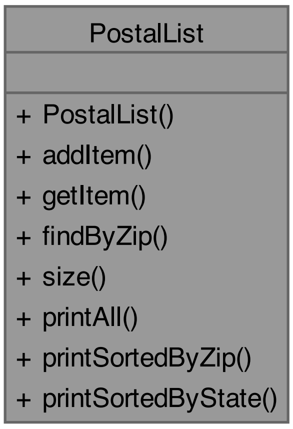
The documentation for this class was generated from the following files:

- source/PostalCodeItem.h
- source/PostalCodeItem.cpp

4.2 PostalList Class Reference

`#include <PostalList.h>`

Collaboration diagram for PostalList:



Public Member Functions

- PostalList ()=default
- void addItem (const PostalCodeItem &item)

    Add a PostalCodeItem to the list.
- PostalCodeItem getItem (int index) const

    Get a PostalCodeItem by index.
- const PostalCodeItem ∗ findByZip (int zip) const

    Find a PostalCodeItem by its ZIP code.
- int size () const

    Get the number of items in the list.
- void printAll () const

    Print all PostalCodeItems in the list.
- void printSortedByZip () const

    Print PostalCodeItems sorted by ZIP code.
- void printSortedByState () const

    Print PostalCodeItems sorted by state and then by ZIP code.

4.2.1 Detailed Description

Definition at line 23 of file PostalList.h.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 PostalList()

```
PostalList::PostalList ()  [default]
```

4.2.3 Member Function Documentation

4.2.3.1 addItem()

```
void PostalList::addItem (
            const PostalCodeItem & item)
```

Add a PostalCodeItem to the list.

Parameters

item	The PostalCodeItem to be added.

Definition at line 26 of file PostalList.cpp.

4.2.3.2 getItem()

```
PostalCodeItem PostalList::getItem (
            int index) const
```

Get a PostalCodeItem by index.

Parameters

index	The index of the item to retrieve.

Returns

The PostalCodeItem at the specified index.

Exceptions

out_of_range	if the index is invalid.

Definition at line 37 of file PostalList.cpp.

4.2.3.3 findByZip()

```
const PostalCodeItem * PostalList::findByZip (
            int zip) const
```

Find a PostalCodeItem by its ZIP code.

Parameters

zip	The ZIP code to search for.

Returns

A pointer to the PostalCodeItem if found, nullptr otherwise.

Note

The returned pointer is valid as long as the PostalList object exists and is not modified.

Definition at line 52 of file PostalList.cpp.

4.2.3.4 size()

```
int PostalList::size () const
```

Get the number of items in the list.

Returns

The number of PostalCodeItem objects in the list.

Definition at line 68 of file PostalList.cpp.

4.2.3.5 printAll()

```
void PostalList::printAll () const
```

Print all PostalCodeItems in the list.

Each item's information is printed followed by a separator line.

Note

The order of items is the same as the order they were added.

Definition at line 78 of file PostalList.cpp.

4.2.3.6 printSortedByZip()

```
void PostalList::printSortedByZip () const
```

Print PostalCodeItems sorted by ZIP code.

Each item's information is printed followed by a separator line.

Note

Items are sorted in ascending order by ZIP code.

Definition at line 92 of file PostalList.cpp.

4.2.3.7 printSortedByState()

```
void PostalList::printSortedByState () const
```

Print PostalCodeItems sorted by state and then by ZIP code.

Each item's information is printed followed by a separator line.

Note

Items are sorted first by state (alphabetically) and then by ZIP code (numerically) within each state.

Definition at line 115 of file PostalList.cpp.

References PostalCodeItem::getState().

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

- source/PostalList.h
- source/PostalList.cpp

Chapter 5

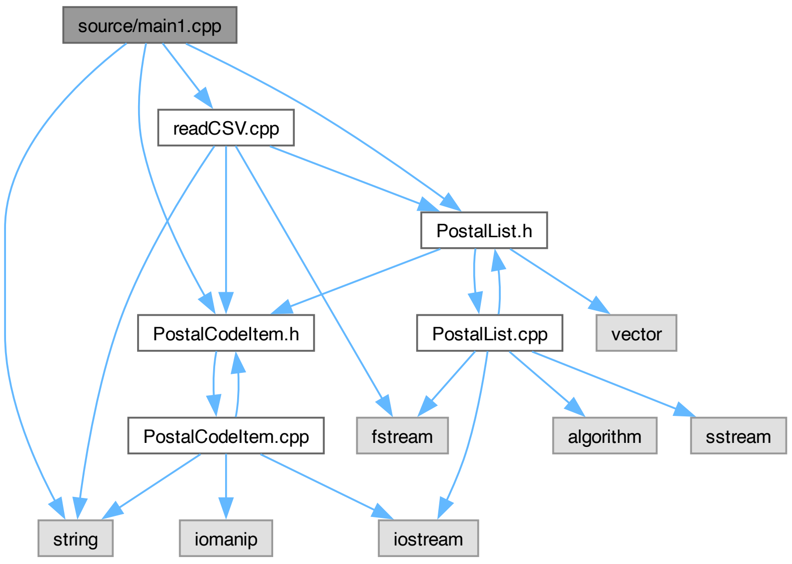
File Documentation

5.1 source/main1.cpp File Reference

Our first main file for "Zip Code Group 4 Project 1.0" that output the postal sorted by zip code.

```
#include <string>
#include "PostalCodeItem.h"
#include "PostalList.h"
#include "readCSV.cpp"
```

Include dependency graph for main1.cpp:



Functions

- int main ()

  Starts the program, loads the CSV, and prints the table.

5.1.1 Detailed Description

Our first main file for "Zip Code Group 4 Project 1.0" that output the postal sorted by zip code.

@course CSCI 331 - Software Systems — Fall 2025 @project Zip Code Group Project 1.0

We read a CSV (made from the XLSX), load each row into our list, and then print the table the assignment asks for: one line per state (A–Z) showing, in this order, the ZIPs that are farthest East (smallest longitude), West (biggest longitude), North (biggest latitude), and South (smallest latitude). We also print a header first.

Authors

- Tran, Minh Quan
- Asfaw, Abel
- Kariniemi, Carson
- Rogers, Mitchell
- Farah, Mahad

Date

Sep 23rd 2025

Version

1.0

Bug None that we know of right now.

Definition in file main1.cpp.

5.1.2 Function Documentation

5.1.2.1 main()

```
int main ()
```

Starts the program, loads the CSV, and prints the table.

This main1.cpp will print the postal sorted by zip code.

Returns

0 if everything went fine.

Precondition

The file `us_postal_codes.csv` is in the same folder and has the expected columns.

Postcondition

We write the header and then one row per state to standard output.

Definition at line 44 of file main1.cpp.

References inputCSVtoList(), and PostalList::printSortedByZip().

5.2 main1.cpp

Go to the documentation of this file.

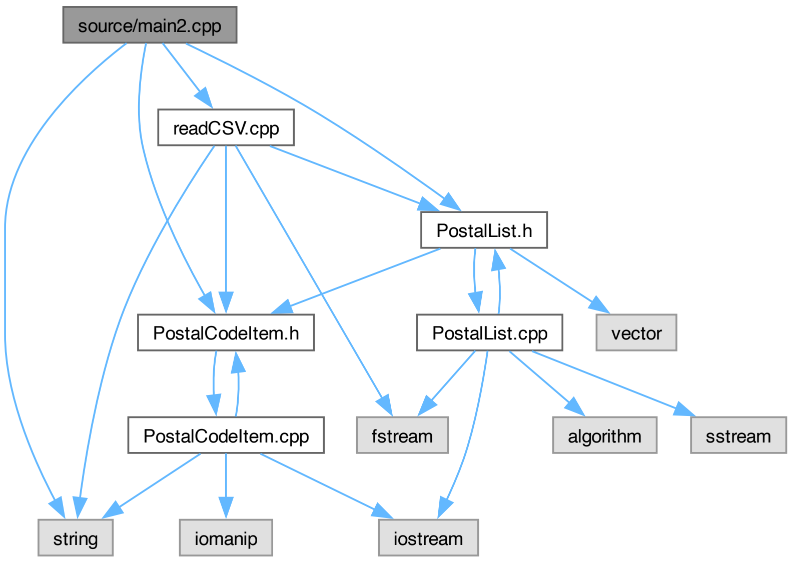
```

#include <string>
#include "PostalCodeItem.h"
#include "PostalList.h"
#include "readCSV.cpp"

using namespace std;

int main()
{
    // Create a variable for csv file name
    string fileName = "us_postal_codes.csv";

    // Create an instance for PostalList
    PostalList myPostalList;

    // Input the data from the CSV file to the postal list
    inputCSVtoList(myPostalList, fileName);

    // Display the header with the appropriate table
    cout « "A table of all the postal sorted by zip:" « endl
         « endl;
    cout « left « setw(10) « "Zip Code"
         « setw(20) « "Place Name"
         « setw(10) « "State"
         « setw(30) « "County"
         « setw(12) « "Latitude"
         « setw(12) « "Longitude"
         « endl;
    cout «
"-----------------------------------------------------------------------------------------------" «
endl;

    // Display the table sorted by zip
    myPostalList.printSortedByZip();

    return 0;
}
```

5.3 source/main2.cpp File Reference

Our second main file for "Zip Code Group 4 Project 1.0" that output the postal sorted by state.

```
#include <string>
#include "PostalCodeItem.h"
#include "PostalList.h"
#include "readCSV.cpp"
```

Include dependency graph for main2.cpp:



Functions

- int main ()

  Starts the program, loads the CSV, and prints the table.

5.3.1 Detailed Description

Our second main file for "Zip Code Group 4 Project 1.0" that output the postal sorted by state.

@course CSCI 331 - Software Systems — Fall 2025 @project Zip Code Group Project 1.0

We read a CSV (made from the XLSX), load each row into our list, and then print the table the assignment asks for: one line per state (A–Z) showing, in this order, the ZIPs that are farthest East (smallest longitude), West (biggest longitude), North (biggest latitude), and South (smallest latitude). We also print a header first.

Authors

- Tran, Minh Quan
- Asfaw, Abel
- Kariniemi, Carson
- Rogers, Mitchell
- Farah, Mahad

Date

Sep 23rd 2025

Version

1.0

Bug None that we know of right now.

Definition in file main2.cpp.

5.3.2 Function Documentation

5.3.2.1 main()

```
int main ()
```

Starts the program, loads the CSV, and prints the table.

This main2.cpp will print the postal sorted by state.

Returns

0 if everything went fine.

Precondition

The file `us_postal_codes.csv` is in the same folder and has the expected columns.

Postcondition

We write the header and then one row per state to standard output.

Definition at line 44 of file main2.cpp.

References inputCSVtoList(), and PostalList::printSortedByState().

5.4 main2.cpp

Go to the documentation of this file.

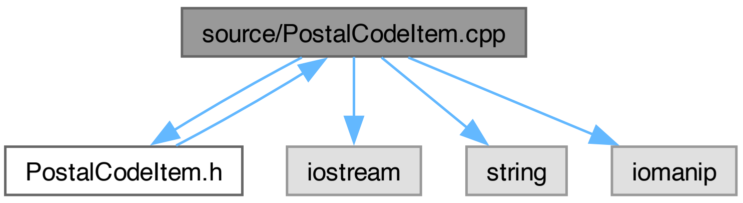
```
00001
00026
00027 #include <string>
00028 #include "PostalCodeItem.h"
00029 #include "PostalList.h"
00030 #include "readCSV.cpp"
00031
00032 using namespace std;
00033
00044 int main()
00045 {
00046     // Create a variable for csv file name
00047     string fileName = "us_postal_codes.csv";
00048
00049     // Create an instance for PostalList
00050     PostalList myPostalList;
00051
00052     // Input the data from the CSV file to the postal list
00053     inputCSVtoList(myPostalList, fileName);
00054
00055     // Display the header with the appropriate table
00056     cout « "A table of all the postal sorted by state:" « endl
00057          « endl;
00058     cout « left « setw(10) « "Zip Code"
00059          « setw(20) « "Place Name"
00060          « setw(10) « "State"
00061          « setw(30) « "County"
00062          « setw(12) « "Latitude"
00063          « setw(12) « "Longitude"
00064          « endl;
00065     cout «
      "------------------------------------------------------------------------------------------------" «
      endl;
00066
00067     // Display the table sorted by state
00068     myPostalList.printSortedByState();
00069
00070     return 0;
00071 }
```

5.5 source/PostalCodeItem.cpp File Reference

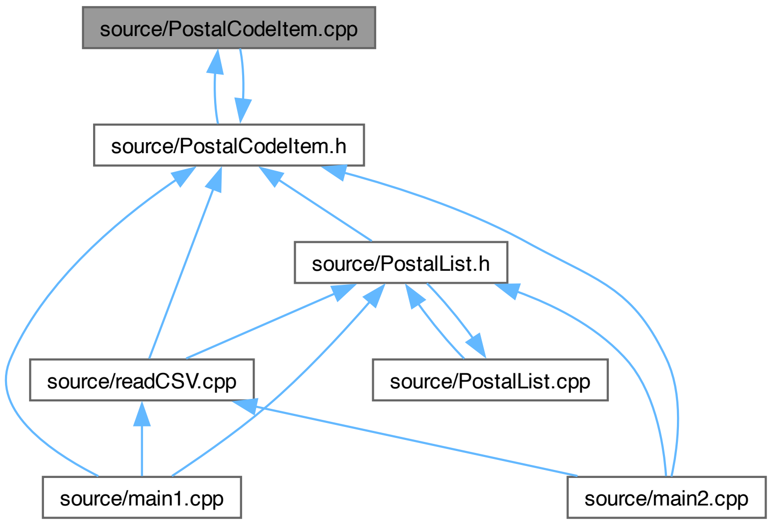
Implementation of the PostalCodeItem class representing a postal code entry.

```
#include "PostalCodeItem.h"
#include <iostream>
#include <string>
#include <iomanip>
```

Include dependency graph for PostalCodeItem.cpp:



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



5.5.1 Detailed Description

Implementation of the PostalCodeItem class representing a postal code entry.

Author

- Asfaw, Abel, Farah, Mahad, Kariniemi, Carson, Rogers, Mitchell Tran, Minh Quan

Version

1.0

Date

2025-9-23

Definition in file PostalCodeItem.cpp.

5.6 PostalCodeItem.cpp

Go to the documentation of this file.

```



#include "PostalCodeItem.h"
#include <iostream>
#include <string>
#include <iomanip>

using namespace std;

PostalCodeItem::PostalCodeItem()
{
    zip = 0;
    place = "";
    state = "";
    county = "";
    latitude = 0;
    longitude = 0;
}

PostalCodeItem::PostalCodeItem(int z, const string &p, const string &s, const string &c, double lat,
      double lon)
{
    zip = z;
    place = p;
    state = s;
    county = c;
    latitude = lat;
    longitude = lon;
}

int PostalCodeItem::getZip() const
{
    return zip;
}

string PostalCodeItem::getPlace() const
{
    return place;
}

string PostalCodeItem::getState() const
{
    return state;
}

string PostalCodeItem::getCounty() const
{
    return county;
}

double PostalCodeItem::getLatitude() const
{
    return latitude;
}

double PostalCodeItem::getLongitude() const
{
    return longitude;
}
```

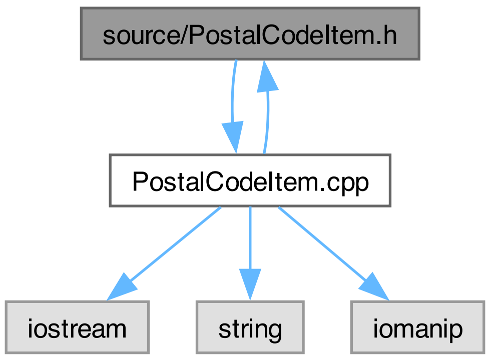
```
void PostalCodeItem::setZip(int newZip)
{
    zip = newZip;
}

void PostalCodeItem::setPlace(const string &newPlace)
{
    place = newPlace;
}

void PostalCodeItem::setState(const string &newState)
{
    state = newState;
}

void PostalCodeItem::setCounty(const string &newCounty)
{
    county = newCounty;
}

void PostalCodeItem::setLatitude(double newLat)
{
    latitude = newLat;
}

void PostalCodeItem::setLongitude(double newLon)
{
    longitude = newLon;
}

void PostalCodeItem::printInfo() const
{
    cout « left « setw(10) « zip
         « setw(20) « place
         « setw(10) « state
         « setw(30) « county
         « setw(12) « latitude
         « setw(12) « longitude
         « endl;
}
```

5.7 source/PostalCodeItem.h File Reference

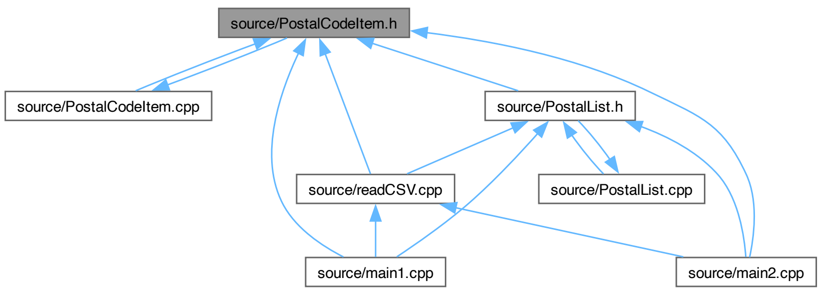
Defines the PostalCodeItem class for representing postal code records.

`#include "PostalCodeItem.cpp"`

Include dependency graph for PostalCodeItem.h:



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



Classes

- class PostalCodeItem

5.7.1 Detailed Description

Defines the PostalCodeItem class for representing postal code records.

Author

Asfaw, Abel, Farah, Mahad, Kariniemi, Carson, Rogers, Mitchell Tran, Minh Quan Each PostalCodeItem stores data about a single postal code including:

- ZIP code
- Place name
- State abbreviation
- County
- Latitude
- Longitude

Definition in file PostalCodeItem.h.

5.8 PostalCodeItem.h

Go to the documentation of this file.

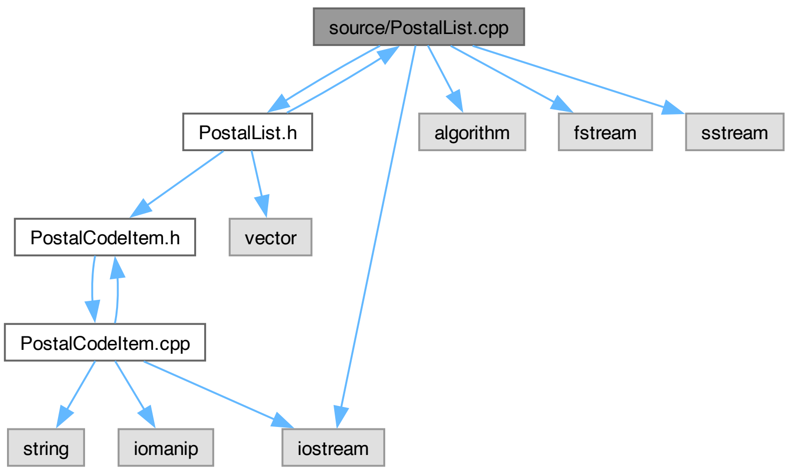
```

#ifndef POSTAL_CODE_ITEM
#define POSTAL_CODE_ITEM

using namespace std;

class PostalCodeItem
{
private:
    int zip;
    string place;
    string state;
    string county;
    double latitude;
    double longitude;

public:
    PostalCodeItem();

    PostalCodeItem(int z, const string &p, const string &s, const string &c, double lat, double lon);

    int getZip() const;

    string getPlace() const;

    string getState() const;

    string getCounty() const;

    double getLatitude() const;

    double getLongitude() const;

    void setZip(int newZip);

    void setPlace(const string &newPlace);

    void setState(const string &newState);

    void setCounty(const string &newCounty);

    void setLatitude(double newLat);

    void setLongitude(double newLon);

    void printInfo() const;
};

#include "PostalCodeItem.cpp"
#endif
```

5.9 source/PostalList.cpp File Reference

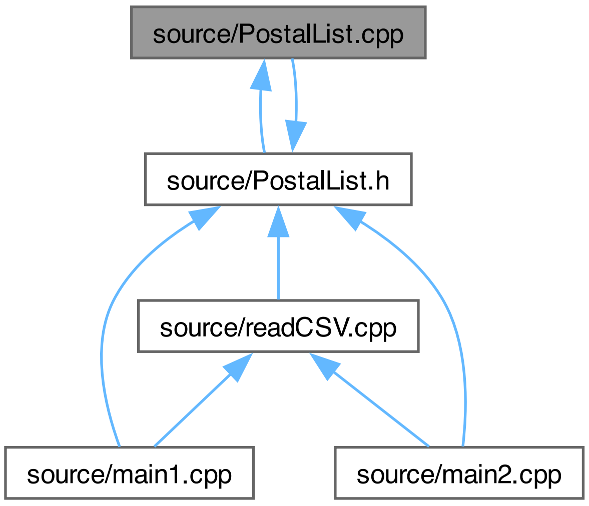
@ brief Implementation of the PostalList class for managing a collection of PostalCodeItem objects.

```
#include "PostalList.h"
#include <iostream>
#include <algorithm>
#include <fstream>
#include <sstream>
```

Include dependency graph for PostalList.cpp:



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



5.9.1 Detailed Description

@ brief Implementation of the PostalList class for managing a collection of PostalCodeItem objects.

@ author Asfaw, Abel, Farah, Mahad, Kariniemi, Carson, Rogers, Mitchell Tran, Minh Quan @ version 1.0 @ date 2025-9-23

Definition in file PostalList.cpp.

5.10 PostalList.cpp

Go to the documentation of this file.

```
#include "PostalList.h"
#include <iostream>
#include <algorithm>
#include <fstream>
#include <sstream>

using namespace std;

void PostalList::addItem(const PostalCodeItem &item)
{
    items.push_back(item);
}

PostalCodeItem PostalList::getItem(int index) const
{
    if (index < items.size())
    {
        return items[index];
    }
    throw out_of_range("Index out of range in PostalList::getItem");
}
```

```

const PostalCodeItem *PostalList::findByZip(int zip) const
{
    for (const auto &item : items)
    {
        if (item.getZip() == zip)
        {
            return &item;
        }
    }
    return nullptr;
}

int PostalList::size() const
{
    return items.size();
}

void PostalList::printAll() const
{
    for (int i = 0; i < items.size(); i++)
    {
        items[i].printInfo();
        cout « "-----------------------------------------------------------------------------------------------" « endl;
    }
}

void PostalList::printSortedByZip() const
{
    // Make a copy so original order is preserved
    vector<PostalCodeItem> sortedItems = items;

    sort(sortedItems.begin(), sortedItems.end(),
         [](const PostalCodeItem &a, const PostalCodeItem &b)
         {
             return a.getZip() < b.getZip();
         });

    for (const auto &item : sortedItems)
    {
        item.printInfo();
        cout « "-----------------------------------------------------------------------------------------------" « endl;
    }
}

void PostalList::printSortedByState() const
{
    // Copy items so we don't change the internal order
    vector<PostalCodeItem> sortedItems = items;

    sort(sortedItems.begin(), sortedItems.end(),
         [](const PostalCodeItem &a, const PostalCodeItem &b)
         {
             if (a.getState() == b.getState())
             {
                 return a.getZip() < b.getZip(); // secondary sort by ZIP
             }
             return a.getState() < b.getState();
         });

    for (const auto &item : sortedItems)
    {
        item.printInfo();
        cout « "-----------------------------------------------------------------------------------------------" « endl;
    }
}
```

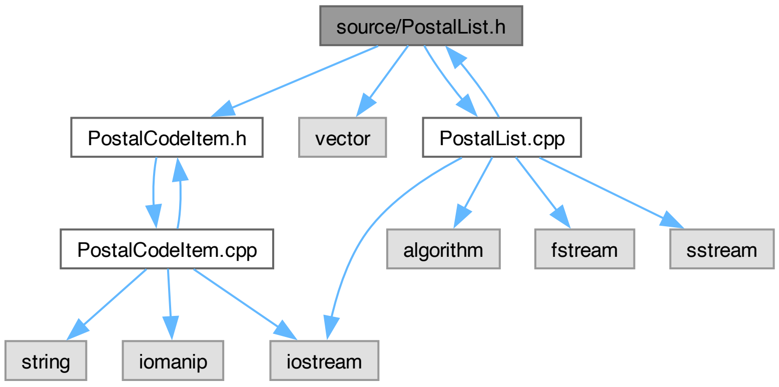
5.11 source/PostalList.h File Reference

Defines the PostalList class for managing collections of postal codes.

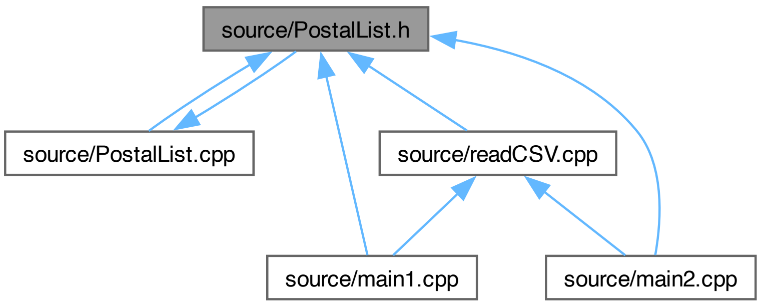
```
#include "PostalCodeItem.h"
#include <vector>
```

```
#include "PostalList.cpp"
```

Include dependency graph for PostalList.h:



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



Classes

- class PostalList

5.11.1 Detailed Description

Defines the PostalList class for managing collections of postal codes.

Author

Asfaw, Abel, Farah, Mahad, Kariniemi, Carson, Rogers, Mitchell Tran, Minh Quan The PostalList class provides storage and utility functions for handling multiple PostalCodeItem objects, including adding, searching, and printing data in sorted order.

Definition in file PostalList.h.

5.12 PostalList.h

Go to the documentation of this file.

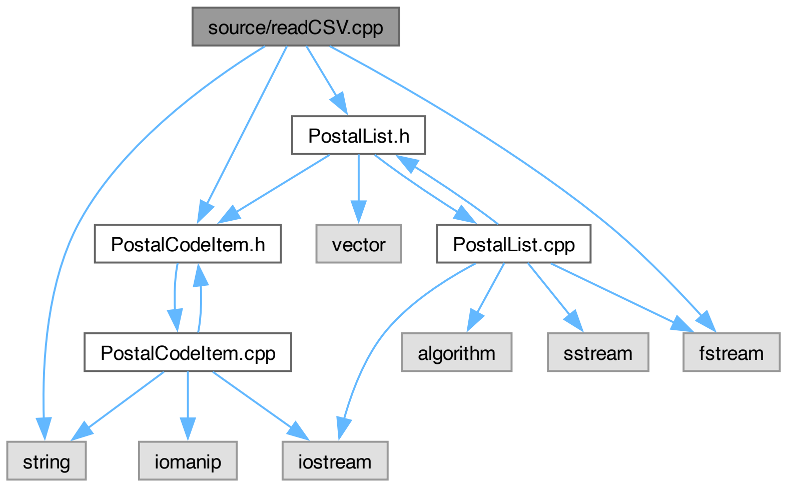
```
#ifndef POSTAL_LIST_H
#define POSTAL_LIST_H

#include "PostalCodeItem.h"
#include <vector>

using namespace std;

class PostalList
{
private:
    vector<PostalCodeItem> items;

public:
    // Constructors
    PostalList() = default;

    void addItem(const PostalCodeItem &item);

    PostalCodeItem getItem(int index) const;

    const PostalCodeItem *findByZip(int zip) const;

    int size() const;

    void printAll() const;

    void printSortedByZip() const;

    void printSortedByState() const;
};

#include "PostalList.cpp"
#endif
```

5.13 source/readCSV.cpp File Reference

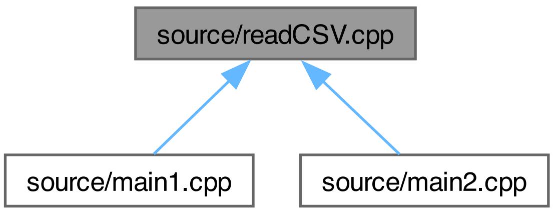
Utility functions for reading postal code data from a CSV file.

```
#include <string>
#include "PostalCodeItem.h"
#include "PostalList.h"
#include <fstream>
```

Include dependency graph for readCSV.cpp:



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



Functions

- void inputCSVtoList (PostalList &inputList, string fileName)

  Reads the CSV and adds each row to the list.

5.13.1 Detailed Description

Utility functions for reading postal code data from a CSV file.

@course CSCI 331 - Software Systems — Fall 2025 @project Zip Code Group Project 1.0

This file defines functions that read U.S. postal code data stored in a CSV file and populate a PostalList object. Each line of the CSV contains one postal record.

Authors

- Tran, Minh Quan
- Asfaw, Abel
- Kariniemi, Carson
- Rogers, Mitchell
- Farah, Mahad

Date

Sep 23rd 2025

Version

1.0

Bug None that we know of right now.

Definition in file readCSV.cpp.

5.13.2 Function Documentation

5.13.2.1 inputCSVtoList()

```
void inputCSVtoList (
            PostalList & inputList,
            string fileName)
```

Reads the CSV and adds each row to the list.

The file has a header, then each line has 6 pieces: ZIP, Place, State, County, Latitude, Longitude

Parameters

inputList	Where we store all the items.
fileName	The CSV file we open.

Precondition

- The CSV exists and we can open it.
- Lines are simple comma-separated (no quotes/commas inside fields).

Postcondition

- Every good line becomes a PostalCodeItem in `inputList`.
- The file is closed before we leave.

Note

We keep this simple on purpose. If the CSV has quotes or weird commas, this version won't handle it.

Definition at line 52 of file readCSV.cpp.

References PostalList::addItem(), PostalCodeItem::setCounty(), PostalCodeItem::setLatitude(), PostalCodeItem::setLongitude(), PostalCodeItem::setPlace(), PostalCodeItem::setState(), and PostalCodeItem::setZip().

5.14 readCSV.cpp

Go to the documentation of this file.

```


#include <string>
#include "PostalCodeItem.h"
#include "PostalList.h"
#include <fstream>

using namespace std;


void inputCSVtoList(PostalList &inputList, string fileName)
{
    PostalCodeItem item;
    string line = "";
    int location = 0;

    ifstream myFile;
    myFile.open(fileName);

    // Skip the header: "zip,place,state,county,latitude,longitude"
    getline(myFile, line);

    while (getline(myFile, line))
    {
        // ZIP
        location = line.find(",");
        item.setZip(stoi(line.substr(0, location)));
        line = line.substr(location + 1, line.length());

        // Place
        location = line.find(",");
        item.setPlace(line.substr(0, location));
        line = line.substr(location + 1, line.length());

        // State
        location = line.find(",");
        item.setState(line.substr(0, location));
        line = line.substr(location + 1, line.length());

        // County
        location = line.find(",");
        item.setCounty(line.substr(0, location));
        line = line.substr(location + 1, line.length());

        // Latitude
        location = line.find(",");
        item.setLatitude(stod(line.substr(0, location)));
        line = line.substr(location + 1, line.length());

        // Longitude (last part of the line)
        item.setLongitude(stod(line));

        // Add it to our list
        inputList.addItem(item);
    }

    myFile.close();
}
```