

Zip Code

Generated by Doxygen 1.9.8

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 ZipCodeBuffer Class Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Function Documentation	5
3.1.2.1 getRecords()	5
3.1.2.2 loadCSV()	5
3.2 ZipCodeProcessor Class Reference	6
3.2.1 Detailed Description	6
3.2.2 Member Function Documentation	6
3.2.2.1 findExtremeZipCodes()	6
3.2.2.2 organizeByState()	6
3.3 ZipCodeRecord Struct Reference	7
3.3.1 Detailed Description	7
4 File Documentation	9
4.1 main.cpp File Reference	9
4.1.1 Detailed Description	9
4.2 zip_code_buffer.h	10
Index	11

Chapter 1

Class Index

1.1 Class List

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

ZipCodeBuffer	Reads and processes a CSV file containing Zip Code records	5
ZipCodeProcessor	Processes and organizes Zip Code data	6
ZipCodeRecord	Stores information about a Zip Code	7

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

main.cpp	Part II: Command-Line Search for Zip Code Records	9
zip_code_buffer.h	10

Chapter 3

Class Documentation

3.1 ZipCodeBuffer Class Reference

Reads and processes a CSV file containing Zip Code records.

```
#include <zip_code_buffer.h>
```

Public Member Functions

- bool [loadCSV](#) (const std::string &filename)
Loads a CSV file containing zip code data into memory.
- std::vector< [ZipCodeRecord](#) > [getRecords](#) () const
Retrieves all stored zip code records.

3.1.1 Detailed Description

Reads and processes a CSV file containing Zip Code records.

3.1.2 Member Function Documentation

3.1.2.1 getRecords()

```
std::vector< ZipCodeRecord > ZipCodeBuffer::getRecords ( ) const
```

Retrieves all stored zip code records.

Returns

A vector of [ZipCodeRecord](#) objects.

3.1.2.2 loadCSV()

```
bool ZipCodeBuffer::loadCSV (  
    const std::string & filename )
```

Loads a CSV file containing zip code data into memory.

Group 5 - Process zip codes from csv file

Parameters

<i>filename</i>	The name of the CSV file to read.
-----------------	-----------------------------------

Returns

True if the file was loaded successfully, otherwise false.

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

- zip_code_buffer.h
- zip_code_buffer.cpp

3.2 ZipCodeProcessor Class Reference

Processes and organizes Zip Code data.

```
#include <zip_code_buffer.h>
```

Public Member Functions

- void [organizeByState](#) (const std::vector< [ZipCodeRecord](#) > &records)
Organizes zip code records by state.
- std::map< std::string, std::tuple< [ZipCodeRecord](#), [ZipCodeRecord](#), [ZipCodeRecord](#), [ZipCodeRecord](#) > > [findExtremeZipCodes](#) () const
Finds the extreme zip codes (east, west, north, south) for each state.
- void [printResults](#) () const
Prints the results in a formatted table.

3.2.1 Detailed Description

Processes and organizes Zip Code data.

3.2.2 Member Function Documentation

3.2.2.1 findExtremeZipCodes()

```
std::map< std::string, std::tuple< ZipCodeRecord, ZipCodeRecord, ZipCodeRecord, ZipCodeRecord > >
ZipCodeProcessor::findExtremeZipCodes ( ) const
```

Finds the extreme zip codes (east, west, north, south) for each state.

Returns

A map where each state is mapped to its corresponding extreme zip codes.

3.2.2.2 organizeByState()

```
void ZipCodeProcessor::organizeByState (
    const std::vector< ZipCodeRecord > & records )
```

Organizes zip code records by state.

Parameters

<i>records</i>	A vector of ZipCodeRecord objects to be categorized by state.
----------------	---

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

- `zip_code_buffer.h`
- `zip_code_buffer.cpp`

3.3 ZipCodeRecord Struct Reference

Stores information about a Zip Code.

```
#include <zip_code_buffer.h>
```

Public Attributes

- `std::string zip_code`
- `std::string place_name`
- `std::string state`
- `std::string county`
- `double latitude`
- `double longitude`

3.3.1 Detailed Description

Stores information about a Zip Code.

Group 5 - Process zip codes from csv file

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

- `zip_code_buffer.h`

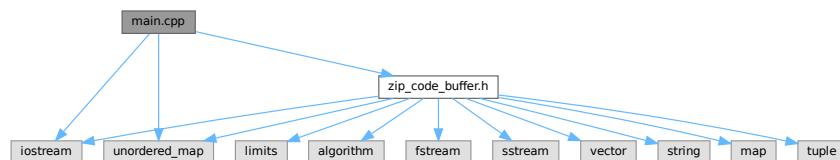
Chapter 4

File Documentation

4.1 main.cpp File Reference

Part II: Command-Line Search for Zip Code Records.

```
#include <iostream>
#include <unordered_map>
#include "zip_code_buffer.h"
Include dependency graph for main.cpp:
```



Functions

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

4.1.1 Detailed Description

Part II: Command-Line Search for Zip Code Records.

This program builds a primary key index from the CSV data file ("zip_codes.csv") and processes command-line flags of the form -Z[ZipCode] to search for specific Zip Code records. For each search, only the targeted record is loaded into memory.

4.2 zip_code_buffer.h

```

00001
00005 #ifndef ZIP_CODE_BUFFER_H
00006 #define ZIP_CODE_BUFFER_H
00007
00008 #include <iostream>
00009 #include <fstream>
00010 #include <sstream>
00011 #include <vector>
00012 #include <string>
00013 #include <map>
00014 #include <tuple>
00015 #include <limits>
00016 #include <algorithm>
00017 #include <unordered_map> // Added for primary key index functionality
00018
00023 struct ZipCodeRecord {
00024     std::string zip_code;
00025     std::string place_name;
00026     std::string state;
00027     std::string county;
00028     double latitude;
00029     double longitude;
00030 };
00031
00036 class ZipCodeBuffer {
00037 private:
00038     std::vector<ZipCodeRecord> records;
00039
00040 public:
00041     bool loadCSV(const std::string& filename);
00042     std::vector<ZipCodeRecord> getRecords() const;
00043 };
00044
00049 class ZipCodeProcessor {
00050 private:
00051     std::map<std::string, std::vector<ZipCodeRecord> > state_map;
00052
00053 public:
00054     void organizeByState(const std::vector<ZipCodeRecord>& records);
00055     std::map<std::string, std::tuple<ZipCodeRecord, ZipCodeRecord, ZipCodeRecord, ZipCodeRecord> >
        findExtremeZipCodes() const;
00056     void printResults() const;
00057 };
00058
00059 // ==== Additional functions for Part II (Command-Line Search) ==== //
00060
00066 std::unordered_map<std::string, std::streampos> buildIndex(const std::string &filename);
00067
00075 ZipCodeRecord readRecordAtOffset(const std::string &filename, std::streampos offset);
00076
00081 void displayRecord(const ZipCodeRecord &record);
00082
00083 #endif // ZIP_CODE_BUFFER_H

```

Index

- findExtremeZipCodes
 - ZipCodeProcessor, [6](#)
- getRecords
 - ZipCodeBuffer, [5](#)
- loadCSV
 - ZipCodeBuffer, [5](#)
- main.cpp, [9](#)
- organizeByState
 - ZipCodeProcessor, [6](#)
- ZipCodeBuffer, [5](#)
 - getRecords, [5](#)
 - loadCSV, [5](#)
- ZipCodeProcessor, [6](#)
 - findExtremeZipCodes, [6](#)
 - organizeByState, [6](#)
- ZipCodeRecord, [7](#)