

PA03

Generated by Doxygen 1.8.12



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	City Class Reference . . . . .	5
3.1.1	Constructor & Destructor Documentation . . . . .	5
3.1.1.1	City() [1/3] . . . . .	5
3.1.1.2	City() [2/3] . . . . .	6
3.1.1.3	City() [3/3] . . . . .	6
3.1.2	Member Function Documentation . . . . .	6
3.1.2.1	addAdjacentCity() . . . . .	6
3.1.2.2	getName() . . . . .	6
3.1.3	Friends And Related Function Documentation . . . . .	6
3.1.3.1	operator<< . . . . .	6
3.2	FlightMap Class Reference . . . . .	7
3.2.1	Constructor & Destructor Documentation . . . . .	7
3.2.1.1	FlightMap() . . . . .	7
3.2.2	Member Function Documentation . . . . .	7
3.2.2.1	getIndexFromName() . . . . .	7
3.2.2.2	getNextCity() . . . . .	7
3.2.2.3	isPath() . . . . .	8
3.2.2.4	markVisited() . . . . .	8
3.2.2.5	unvisitAll() . . . . .	8
3.3	Stack< T > Class Template Reference . . . . .	8
3.3.1	Member Function Documentation . . . . .	9
3.3.1.1	isEmpty() . . . . .	9
3.3.1.2	peek() . . . . .	9
3.3.1.3	pop() . . . . .	9
3.3.1.4	push() . . . . .	9

<b>4 File Documentation</b>	<b>11</b>
4.1 City.cpp File Reference . . . . .	11
4.1.1 Detailed Description . . . . .	11
4.1.2 Function Documentation . . . . .	11
4.1.2.1 operator<<() . . . . .	11
4.2 City.h File Reference . . . . .	12
4.2.1 Detailed Description . . . . .	12
4.3 FlightMap.v1.cpp File Reference . . . . .	12
4.3.1 Detailed Description . . . . .	12
4.4 PA03.cpp File Reference . . . . .	13
4.4.1 Detailed Description . . . . .	13
4.5 Stack.cpp File Reference . . . . .	13
4.5.1 Detailed Description . . . . .	13
<b>Index</b>	<b>15</b>

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">City</a>	5
<a href="#">FlightMap</a>	7
<a href="#">Stack&lt; T &gt;</a>	8



## Chapter 2

# File Index

### 2.1 File List

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

<a href="#">City.cpp</a>	Implementation file for the <a href="#">City</a> Class . . . . .	11
<a href="#">City.h</a>	Header file for the <a href="#">City</a> Class . . . . .	12
<a href="#">FlightMap.v1.cpp</a>	Implementation file for the <a href="#">FlightMap</a> Class . . . . .	12
<b>FlightMap.v1.h</b>	. . . . .	??
<a href="#">PA03.cpp</a>	Implementation file for the the main driver . . . . .	13
<a href="#">Stack.cpp</a>	Header/Implementation file for the <a href="#">Stack</a> Class . . . . .	13





## Chapter 3

# Class Documentation

### 3.1 City Class Reference

#### Public Member Functions

- `City ()`  
*Default constructor for the class.*
- `City (string)`  
*Custom constructor for the class.*
- `City (const City &toCopy)`  
*Copy constructor for the class.*
- `void addAdjacentCity (City adjacent)`  
*addAdjacentCity function*
- `string getName () const`  
*getName function*

#### Public Attributes

- `bool visited`
- `Stack< City > adjacentCities`

#### Friends

- `ostream & operator<< (ostream &out, const City &city)`  
*ostream << overloader*

#### 3.1.1 Constructor & Destructor Documentation

##### 3.1.1.1 `City()` [1/3]

```
City::City ( )
```

Default constructor for the class.

constructs the class

#### 3.1.1.2 City() [2/3]

```
City::City (
    string cityName )
```

Custom constructor for the class.

constructs the class from the name of a city

#### 3.1.1.3 City() [3/3]

```
City::City (
    const City & toCopy )
```

Copy constructor for the class.

constructs the class from a copy

### 3.1.2 Member Function Documentation

#### 3.1.2.1 addAdjacentCity()

```
void City::addAdjacentCity (
    City adjacent )
```

addAdjacentCity function

pushes a city to the adjacentCities stack

#### 3.1.2.2 getName()

```
string City::getName ( ) const
```

getName function

returns the name of the city

### 3.1.3 Friends And Related Function Documentation

#### 3.1.3.1 operator<<

```
ostream& operator<< (
    ostream & out,
    const City & city ) [friend]
```

ostream << overloader

enables printing with cout

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

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

## 3.2 FlightMap Class Reference

### Public Member Functions

- [FlightMap](#) ([City](#) cities[100], int cityCount)  
*Default constructor for the class.*
- int [getIndexFromName](#) (string name)  
*getIndexFromName function*
- void [markVisited](#) ([City](#) &aCity)  
*markVisited function*
- void [unvisitAll](#) ()  
*unvisitAll function*
- [City](#) [getNextCity](#) ([City](#) fromCity)  
*getNextCity function*
- bool [isPath](#) (string originName, string destinationName)  
*isPath function*

### 3.2.1 Constructor & Destructor Documentation

#### 3.2.1.1 FlightMap()

```
FlightMap::FlightMap (  
    City cities[100],  
    int cityCount )
```

Default constructor for the class.

constructs the class

### 3.2.2 Member Function Documentation

#### 3.2.2.1 getIndexFromName()

```
int FlightMap::getIndexFromName (  
    string name )
```

getIndexFromName function

returns the index of the city, given a name

#### 3.2.2.2 getNextCity()

```
City FlightMap::getNextCity (  
    City fromCity )
```

getNextCity function

(supposed to return the next (unvisited) city)

### 3.2.2.3 isPath()

```
bool FlightMap::isPath (
    string originName,
    string destinationName )
```

isPath function

(supposed to return a true if a destination can be visited, false otherwise)

### 3.2.2.4 markVisited()

```
void FlightMap::markVisited (
    City & aCity )
```

markVisited function

sets a city's visited boolean as true

### 3.2.2.5 unvisitAll()

```
void FlightMap::unvisitAll ( )
```

unvisitAll function

iterates through cities array and marks all as unvisited

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

- FlightMap.v1.h
- [FlightMap.v1.cpp](#)

## 3.3 Stack< T > Class Template Reference

### Public Member Functions

- bool [isEmpty](#) ()  
*isEmpty function*
- void [push](#) (T item)  
*push function*
- T [peek](#) ()  
*peek function*
- void [pop](#) ()  
*pop function*

### 3.3.1 Member Function Documentation

#### 3.3.1.1 isEmpty()

```
template<class T >
bool Stack< T >::isEmpty ( )
```

isEmpty function

returns if the stack is empty or not

#### 3.3.1.2 peek()

```
template<class T >
T Stack< T >::peek ( )
```

peek function

returns item at the top of the stack

#### 3.3.1.3 pop()

```
template<class T >
void Stack< T >::pop ( )
```

pop function

removes the item at the top of the stack

#### 3.3.1.4 push()

```
template<class T>
void Stack< T >::push (
    T item )
```

push function

adds an item to the stack

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

- [Stack.cpp](#)



## Chapter 4

# File Documentation

### 4.1 City.cpp File Reference

Implementation file for the [City](#) Class.

```
#include "City.h"
```

#### Functions

- ostream & [operator<<](#) (ostream &out, const [City](#) &city)  
*ostream << overloader*

#### 4.1.1 Detailed Description

Implementation file for the [City](#) Class.

##### Author

Willis Allstead

Implements functions for the [City](#) Class

##### Version

0.50

#### 4.1.2 Function Documentation

##### 4.1.2.1 [operator<<\(\)](#)

```
ostream& operator<< (  
    ostream & out,  
    const City & city )
```

[ostream << overloader](#)

enables printing with cout

## 4.2 City.h File Reference

Header file for the [City](#) Class.

```
#include <iostream>
#include "Stack.cpp"
```

### Classes

- class [City](#)

### 4.2.1 Detailed Description

Header file for the [City](#) Class.

#### Author

Willis Allstead

Defines functions for the [City](#) Class

#### Version

0.50

## 4.3 FlightMap.v1.cpp File Reference

Implementation file for the [FlightMap](#) Class.

```
#include "FlightMap.v1.h"
```

### 4.3.1 Detailed Description

Implementation file for the [FlightMap](#) Class.

#### Author

Willis Allstead

Implements functions for the [FlightMap](#) Class

#### Version

0.50



## 4.4 PA03.cpp File Reference

Implementation file for the the main driver.

```
#include <iostream>
#include <string>
#include <fstream>
#include "Stack.cpp"
#include "City.h"
#include "FlightMap.v1.h"
```

### Functions

- int **main** ()

#### 4.4.1 Detailed Description

Implementation file for the the main driver.

##### Author

Willis Allstead

Runs all the code in the project

##### Version

0.50

## 4.5 Stack.cpp File Reference

Header/Implementation file for the [Stack](#) Class.

```
#include <iostream>
#include <stdexcept>
#include <vector>
```

### Classes

- class [Stack](#)< T >

#### 4.5.1 Detailed Description

Header/Implementation file for the [Stack](#) Class.

##### Author

Willis Allstead

Defines functions for the [Stack](#) Class

##### Version

0.50



# Index

- addAdjacentCity
  - City, [6](#)
- City, [5](#)
  - addAdjacentCity, [6](#)
  - City, [5](#), [6](#)
  - getName, [6](#)
  - operator<<, [6](#)
- City.cpp, [11](#)
  - operator<<, [11](#)
- City.h, [12](#)
- FlightMap, [7](#)
  - FlightMap, [7](#)
  - getIndexFromName, [7](#)
  - getNextCity, [7](#)
  - isPath, [7](#)
  - markVisited, [8](#)
  - unvisitAll, [8](#)
- FlightMap.v1.cpp, [12](#)
- getIndexFromName
  - FlightMap, [7](#)
- getName
  - City, [6](#)
- getNextCity
  - FlightMap, [7](#)
- isEmpty
  - Stack, [9](#)
- isPath
  - FlightMap, [7](#)
- markVisited
  - FlightMap, [8](#)
- operator<<
  - City, [6](#)
  - City.cpp, [11](#)
- PA03.cpp, [13](#)
- peek
  - Stack, [9](#)
- pop
  - Stack, [9](#)
- push
  - Stack, [9](#)
- Stack
  - isEmpty, [9](#)
  - peek, [9](#)
  - pop, [9](#)
  - push, [9](#)
  - Stack< T >, [8](#)
  - Stack.cpp, [13](#)
  - unvisitAll
    - FlightMap, [8](#)