

Package Deilvery

Generated by Doxygen 1.8.11

Contents

1	CS3528_Project	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	Client Class Reference	5
3.1.1	Detailed Description	6
3.1.2	Constructor & Destructor Documentation	6
3.1.2.1	Client(const string &, const string &)	6
3.1.3	Member Function Documentation	6
3.1.3.1	getAddress() const	6
3.1.3.2	getName() const	6
3.1.3.3	getReceivedPackages() const	6
3.1.3.4	getSentPackages() const	7
3.1.3.5	receivePackage(Package *pack)	7
3.1.3.6	sendPackage(Package *pack)	7
3.1.3.7	setAddress(const string &)	7
3.1.3.8	setName(const string &)	7
3.1.3.9	toString() const	7
3.2	Package Class Reference	8
3.2.1	Detailed Description	9
3.2.2	Constructor & Destructor Documentation	9
3.2.2.1	Package(Client *, Client *, const double, const Priority)	9
3.2.3	Member Function Documentation	9
3.2.3.1	getPriority() const	9
3.2.3.2	getPriorityString() const	9
3.2.3.3	getReceiver() const	9
3.2.3.4	getSender() const	10
3.2.3.5	getWeight() const	10
3.2.3.6	setPriority(const Priority)	10
3.2.3.7	setReceiver(Client *)	10
3.2.3.8	setSender(Client *)	10
3.2.3.9	setWeight(const double)	11

Index	13
-----------------------	----

Chapter 1

CS3528_Project

Final project for class CS3528

Chapter 2

Class Index

2.1 Class List

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

Client		
Client	Client class	5
Package		
Package	Package class	8

Chapter 3

Class Documentation

3.1 Client Class Reference

[Client](#) class.

```
#include <Client.h>
```

Public Member Functions

- [Client](#) (const string &, const string &)
[Client](#) basic constructor.
- void [setName](#) (const string &)
Setter for name.
- string [getName](#) () const
Getter for name.
- void [setAddress](#) (const string &)
Setter for address.
- string [getAddress](#) () const
Getter for address.
- void [sendPackage](#) ([Package](#) *pack)
Add a sent package.
- vector< [Package](#) * > [getSentPackages](#) () const
Get sent packages.
- void [receivePackage](#) ([Package](#) *pack)
Add a received package.
- vector< [Package](#) * > [getReceivedPackages](#) () const
Get received packages.
- string [toString](#) () const
To String.

Friends

- std::ostream & [operator<<](#) (std::ostream &, const [Client](#) &)
Insertion Operator override.

3.1.1 Detailed Description

[Client](#) class.

This client class defines a citizen of the town who can send and receive packages. Each person has a name, address, and vectors representing records of all packages sent and received.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 `Client::Client (const string & name, const string & address)`

[Client](#) basic constructor.

Constructor.

Basic constructor takes strings for name and address.

Parameters

<i>&name</i>	reference to client's name
<i>&address</i>	reference to client's address

3.1.3 Member Function Documentation

3.1.3.1 `string Client::getAddress () const`

Getter for address.

Returns

Street Address

3.1.3.2 `string Client::getName () const`

Getter for name.

Returns

name

3.1.3.3 `vector< Package * > Client::getReceivedPackages () const`

Get received packages.

Returns

Vector of received packages

3.1.3.4 `vector< Package * > Client::getSentPackages () const`

Get sent packages.

Returns

Vector of sent packages

3.1.3.5 `void Client::receivePackage (Package * pack)`

Add a received package.

Parameters

<i>pack</i>	pointer to package
-------------	--------------------

3.1.3.6 `void Client::sendPackage (Package * pack)`

Add a sent package.

Parameters

<i>pack</i>	pointer to package
-------------	--------------------

3.1.3.7 `void Client::setAddress (const string & a)`

Setter for address.

Parameters

<i>a</i>	reference to address string
----------	-----------------------------

3.1.3.8 `void Client::setName (const string & n)`

Setter for name.

Parameters

<i>n</i>	reference to name string
----------	--------------------------

3.1.3.9 `string Client::toString () const`

To String.

Prints out client's name and address and also all packages associated with the client. Insertion operator only prints name and address.

Returns

string representation of client and all sent and received packages

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

- Client.h
- Client.cpp

3.2 Package Class Reference

[Package](#) class.

```
#include <Package.h>
```

Public Types

- enum **Priority** { **REGULAR**, **TWO_DAY**, **OVERNIGHT** }

Public Member Functions

- [Package](#) ([Client](#) *, [Client](#) *, const double, const Priority)
Constructor.
- void [setSender](#) ([Client](#) *)
Sender setter.
- [Client](#) * [getSender](#) () const
Sender getter.
- void [setReceiver](#) ([Client](#) *)
Receiver setter.
- [Client](#) * [getReceiver](#) () const
Receiver getter.
- void [setWeight](#) (const double)
Weight setter.
- double [getWeight](#) () const
Weight getter.
- void [setPriority](#) (const Priority)
Priority setter.
- Priority [getPriority](#) () const
Priority getter.
- string [getPriorityString](#) () const
Priority string getter.

Friends

- std::ostream & [operator<<](#) (std::ostream &, const [Package](#) &)
Insertion Operator Override.

3.2.1 Detailed Description

[Package](#) class.

This [Package](#) class defines a package which was sent by a [Client](#) and received by a [Client](#). Each package has a priority, a weight, and two clients (a sender and receiver).

3.2.2 Constructor & Destructor Documentation

3.2.2.1 `Package::Package (Client * s, Client * r, const double w, const Priority p)`

Constructor.

Parameters

<i>s</i>	Sender Client
<i>r</i>	Receiver Client
<i>w</i>	Package weight
<i>p</i>	Package priority

3.2.3 Member Function Documentation

3.2.3.1 `Package::Priority Package::getPriority () const`

Priority getter.

[Package](#) priority getter.

Returns

[Package](#) priority

3.2.3.2 `string Package::getPriorityString () const`

Priority string getter.

Priority is an enum and hence is actually an int. Here's a nice way to print the string value.

Returns

string of priority

3.2.3.3 `Client * Package::getReceiver () const`

Receiver getter.

Returns

pointer to receiver client

3.2.3.4 `Client * Package::getSender () const`

Sender getter.

Returns

pointer to sender client

3.2.3.5 `double Package::getWeight () const`

Weight getter.

[Package](#) Weight getter.

Returns

[Package](#) weight

3.2.3.6 `void Package::setPriority (const Priority p)`

Priority setter.

[Package](#) Priority setter.

Parameters

<i>p</i>	Priority enum
----------	---------------

3.2.3.7 `void Package::setReceiver (Client * r)`

Receiver setter.

Parameters

<i>r</i>	pointer to receiver client
----------	----------------------------

3.2.3.8 `void Package::setSender (Client * s)`

Sender setter.

Parameters

<i>s</i>	pointer to sender client
----------	--------------------------

3.2.3.9 void Package::setWeight (const double w)

Weight setter.

[Package](#) weight setter.

Throws error on negative weight.

Parameters

<i>w</i>	Package weight
----------	--------------------------------

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

- Package.h
- Package.cpp

Index

Client, [5](#)

 Client, [6](#)

 getAddress, [6](#)

 getName, [6](#)

 getReceivedPackages, [6](#)

 getSentPackages, [6](#)

 receivePackage, [7](#)

 sendPackage, [7](#)

 setAddress, [7](#)

 setName, [7](#)

 toString, [7](#)

getAddress

 Client, [6](#)

getName

 Client, [6](#)

getPriority

 Package, [9](#)

getPriorityString

 Package, [9](#)

getReceivedPackages

 Client, [6](#)

getReceiver

 Package, [9](#)

getSender

 Package, [9](#)

getSentPackages

 Client, [6](#)

getWeight

 Package, [10](#)

Package, [8](#)

 getPriority, [9](#)

 getPriorityString, [9](#)

 getReceiver, [9](#)

 getSender, [9](#)

 getWeight, [10](#)

 Package, [9](#)

 setPriority, [10](#)

 setReceiver, [10](#)

 setSender, [10](#)

 setWeight, [10](#)

receivePackage

 Client, [7](#)

sendPackage

 Client, [7](#)

setAddress

 Client, [7](#)

setName

 Client, [7](#)

setPriority

 Package, [10](#)

setReceiver

 Package, [10](#)

setSender

 Package, [10](#)

setWeight

 Package, [10](#)

toString

 Client, [7](#)