

Stack

1

Generated by Doxygen 1.8.18

1 Class Index	1
1 Class Index	1
1.1 Class List	1
2 File Index	1
2.1 File List	1
3 Class Documentation	2
3.1 Stack::Node Class Reference	2
3.1.1 Constructor & Destructor Documentation	2
3.1.2 Member Data Documentation	2
3.2 Stack Class Reference	2
3.2.1 Detailed Description	3
3.2.2 Member Typedef Documentation	3
3.2.3 Constructor & Destructor Documentation	3
3.2.4 Member Function Documentation	4
3.2.5 Member Data Documentation	5
4 File Documentation	6
4.1 driver.cpp File Reference	6
4.1.1 Function Documentation	6
4.2 Stack.cpp File Reference	6
4.3 Stack.h File Reference	6
4.3.1 Macro Definition Documentation	7
4.3.2 Typedef Documentation	7
Index	9

1 Class Index

1.1 Class List

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

Stack::Node	2
Stack	
Class for managing stack items	2

2 File Index

2.1 File List

Here is a list of all files with brief descriptions:

driver.cpp	6
Stack.cpp	6
Stack.h	6

3 Class Documentation

3.1 Stack::Node Class Reference

Public Member Functions

- **Node** (**StackElement** value, **Node** *link=0)

Public Attributes

- **StackElement** data
- **Node** * next

3.1.1 Constructor & Destructor Documentation

3.1.1.1 Node() `Stack::Node::Node (
StackElement value,
Node * link = 0) [inline]`

3.1.2 Member Data Documentation

3.1.2.1 data `StackElement Stack::Node::data`

3.1.2.2 next `Node* Stack::Node::next`

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

- **Stack.h**

3.2 Stack Class Reference

Class for managing stack items.

```
#include <Stack.h>
```

Classes

- class **Node**

Public Member Functions

- **Stack** ()
Constructs a new instance.
- **Stack** (const **Stack** &original)
Constructs a new instance.
- **~Stack** ()
Destroys the object.
- const **Stack** & **operator=** (const **Stack** &rightHandSide)
Assignment operator.
- bool **empty** () const
Check to see if the stack is empty.
- void **push** (const **StackElement** &value)
Push an object onto the stack.
- void **display** (ostream &out) const
Displays the stack.
- **StackElement** **top** () const
returns the top element of the stack
- void **pop** ()
Pops the object off the stack.

Private Types

- typedef **Node** * **NodePointer**

Private Attributes

- **NodePointer** myTop

3.2.1 Detailed Description

Class for managing stack items.

3.2.2 Member Typedef Documentation

3.2.2.1 **NodePointer** typedef **Node*** **Stack::NodePointer** [private]

3.2.3 Constructor & Destructor Documentation

3.2.3.1 Stack() [1/2] `Stack::Stack ()`

Constructs a new instance.

3.2.3.2 Stack() [2/2] `Stack::Stack (
const Stack & original)`

Constructs a new instance.

Parameters

<i>in</i>	<i>original</i>	The value to add to the stack
-----------	-----------------	-------------------------------

3.2.3.3 ~Stack() `Stack::~~Stack ()`

Destroys the object.

3.2.4 Member Function Documentation**3.2.4.1 display()** `void Stack::display (
ostream & out) const`

Displays the stack.

Parameters

<i>out</i>	stdio object to chain to cout
------------	-------------------------------

3.2.4.2 empty() `bool Stack::empty () const`

Check to see if the stack is empty.

Returns

true if empty, false if it contains elements

3.2.4.3 operator=() `const Stack & Stack::operator= (
const Stack & rightHandSide)`

Assignment operator.

Parameters

in	<i>rightHandSide</i>	The right hand side
----	----------------------	---------------------

Returns

The result of the assignment

3.2.4.4 pop() `void Stack::pop ()`

Pops the object off the stack.

Removes the object on the top of the stack.

3.2.4.5 push() `void Stack::push (`
`const StackElement & value)`

Push an object onto the stack.

Parameters

in	<i>value</i>	The value to add to the stack
----	--------------	-------------------------------

3.2.4.6 top() `StackElement Stack::top () const`

returns the top element of the stack

Returns

The stack element at top

3.2.5 Member Data Documentation**3.2.5.1 myTop** `NodePointer Stack::myTop [private]`

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

- **Stack.h**
- **Stack.cpp**

4 File Documentation

4.1 driver.cpp File Reference

```
#include <iostream>
#include "Stack.h"
#include <cstdio>
```

Functions

- `int main (void)`
Main function is rain here for stack.

4.1.1 Function Documentation

4.1.1.1 main() `int main (`
`void)`

Main function is rain here for stack.

Returns

returns a zero upon completion of given code

4.2 Stack.cpp File Reference

```
#include "stack.h"
#include <new>
#include <iostream>
```

4.3 Stack.h File Reference

```
#include <iostream>
```

Classes

- class **Stack**
Class for managing stack items.
- class **Stack::Node**

Macros

- `#define STACK`

Typedefs

- `typedef int StackElement`

4.3.1 Macro Definition Documentation

4.3.1.1 STACK `#define STACK`

4.3.2 Typedef Documentation

4.3.2.1 StackElement `typedef int StackElement`

Index

- ~Stack
 - Stack, 4
- data
 - Stack::Node, 2
- display
 - Stack, 4
- driver.cpp, 6
 - main, 6
- empty
 - Stack, 4
- main
 - driver.cpp, 6
- myTop
 - Stack, 5
- next
 - Stack::Node, 2
- Node
 - Stack::Node, 2
- NodePointer
 - Stack, 3
- operator=
 - Stack, 4
- pop
 - Stack, 5
- push
 - Stack, 5
- STACK
 - Stack.h, 7
- Stack, 2
 - ~Stack, 4
 - display, 4
 - empty, 4
 - myTop, 5
 - NodePointer, 3
 - operator=, 4
 - pop, 5
 - push, 5
 - Stack, 3, 4
 - top, 5
- Stack.cpp, 6
- Stack.h, 6
 - STACK, 7
 - StackElement, 7
- Stack::Node, 2
 - data, 2
 - next, 2
 - Node, 2
- StackElement
 - Stack.h, 7
- top
 - Stack, 5