My Project

Generated by Doxygen 1.8.17

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

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

Chapter 3

Class Documentation

3.1 Array Class Reference

Public Member Functions

- Array (int=10)
- Array (const Array &)
- ∼Array ()
- int getSize () const
- const Array & operator= (const Array &)
- bool operator== (const Array &) const
- bool operator!= (const Array &) const
- int & operator[ ] (int)

Static Public Member Functions

- static int getArrayCount ()

Friends

- istream & operator>> (istream &, Array &)
- ostream & operator<< (ostream &, const Array &)

3.1.1 Constructor & Destructor Documentation

3.1.1.1 Array() [1/2]

```
Array::Array (
            int arraySize = 10 )
```

Default constructor

Parameters

arraySize	The number of elements the array will hold.

Precondition

None

Postcondition

ptr points to an array of size arraySize and all elements of the array have been initialized to zero. arrayCount is incremented. A negative arraySize results in an array of size 10.

3.1.1.2 Array() [2/2]

```
Array::Array (
         const Array & init )
```

Copy constructor

Parameters

init	

Precondition

init.ptr points to an array of size at least init.size

Postcondition

init is copied into ∗this, arrayCount is incremented

3.1.1.3 ∼Array()

```
Array::∼Array ( )
```

Destructor

Precondition

ptr points to memory on the heap

Postcondition

Array for ptr is deallocated, arrayCount is decremented

3.1.2 Member Function Documentation

3.1.2.1 getArrayCount()

```
int Array::getArrayCount ( )  [static]
```

description - returns arrayCount

Precondition

None

Postcondition

returns the number of arrays

Returns

3.1.2.2 getSize()

```
int Array::getSize ( ) const
```

getSize

Precondition

None

Postcondition

Returns the size of the array

Returns

The size of the array

3.1.2.3 operator"!=()

```
bool Array::operator!= (
             const Array & right ) const
```

Not equal operator - description

Parameters

right	

Precondition

ptr and right.ptr point to arrays with size at least size and size.right

Postcondition

false is returned if the arrays have the same size and elements true is returned otherwise

Returns

3.1.2.4 operator=()

```
const Array & Array::operator= (
             const Array & right )
```

Copy assignment

Parameters

right	

Precondition

right.ptr points to an array of size at least right.size

Postcondition

∗this is assigned the same array as right

Returns

???

3.1.2.5 operator==()

```
bool Array::operator== (
             const Array & right ) const
```

Determines if two arrays are equal.

Parameters

right	The right hand side array to be compared.

Precondition

ptr and right.ptr point to arrays with size at least size and right.size.

Postcondition

Returns

True if the arrays have the same size and elements, false otherwise.

3.1.2.6 operator[]()

```
int & Array::operator[] (
             int subscript )
```

Array access operator - description.

Parameters

subscript	

Precondition

0 <= subscript < size

Postcondition

returns the array value at position "subscript"

Returns

3.1.3 Friends And Related Function Documentation

3.1.3.1 operator<<

```
ostream& operator<< (
             ostream & output,
             const Array & a )   [friend]
```

Extraction operator.

Parameters

istream	something about the istream
array	something about array

Precondition

Postcondition

Extraction operator - description

Parameters

output	
a	

Precondition

> a.ptr must point to an array with size at least a.size

Postcondition

> The first a.size elements of a.ptr are sent to the output istream 10 per line with a trailing endl

Returns

3.1.3.2 operator>>

```
istream& operator>> (
            istream & input,
            Array & a )  [friend]
```

Insertion operator.

Parameters

istream	something about the istream
array	something about array

Precondition

Postcondition

Insertion operator - description

Parameters

input	
a	

Precondition

a.ptr must point to an array with size at least a.size

Postcondition

The first a.size elements of a.ptr are filled with integers read from the input stream.

Returns

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

- array.h
- array.cpp

Chapter 4

File Documentation

4.1 array.cpp File Reference

```
#include "array.h"
#include <cassert>
```

Functions

- istream & operator>> (istream &input, Array &a)
- ostream & operator<< (ostream &output, const Array &a)

4.1.1 Detailed Description

Author

cal

4.1.2 Function Documentation

4.1.2.1 operator<<()

```
ostream& operator<< (
            ostream & output,
            const Array & a )
```

Extraction operator - description

Parameters

output	
a	

Precondition

a.ptr must point to an array with size at least a.size

Postcondition

The first a.size elements of a.ptr are sent to the output istream 10 per line with a trailing endl

Returns

4.1.2.2 operator>>()

```
istream& operator>> (
             istream & input,
             Array & a )
```

Insertion operator - description

Parameters

input	
a	

Precondition

a.ptr must point to an array with size at least a.size

Postcondition

The first a.size elements of a.ptr are filled with integers read from the input stream.

Returns

4.2 array.h File Reference

```
#include <iostream>
```

Classes

- class Array

4.2.1 Detailed Description

Author

cal

Index