

Lab 06

Generated by Doxygen 1.8.11

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	ExprTree< DataType > Class Template Reference	5
3.1.1	Constructor & Destructor Documentation	5
3.1.1.1	ExprTree()	5
3.1.1.2	ExprTree(const ExprTree &source)	6
3.1.1.3	~ExprTree()	6
3.1.2	Member Function Documentation	7
3.1.2.1	build()	7
3.1.2.2	clear()	7
3.1.2.3	commute()	8
3.1.2.4	evaluate() const	8
3.1.2.5	expression() const	8
3.1.2.6	isEquivalent(const ExprTree &source) const	9
3.1.2.7	operator=(const ExprTree &source)	9
3.1.2.8	showStructure() const	9
4	File Documentation	11
4.1	ExpressionTree.cpp File Reference	11
4.1.1	Detailed Description	11
4.2	ExpressionTree.h File Reference	11
	Index	13

Chapter 1

Class Index

1.1 Class List

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

ExprTree< DataType >	5
------------------------------------------------------	---

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

ExpressionTree.cpp	
An Implementation file for an Expression Tree	11
ExpressionTree.h	11

Chapter 3

Class Documentation

3.1 ExprTree< DataType > Class Template Reference

```
#include <ExpressionTree.h>
```

Public Member Functions

- [ExprTree](#) ()
default constructor for ExprTree class
- [ExprTree](#) (const [ExprTree](#) &source)
default copy constructor for ExprTree class
- [ExprTree](#) & [operator=](#) (const [ExprTree](#) &source)
overloaded operator for ExprTree class
- [~ExprTree](#) ()
default destructor for ExprTree class
- void [build](#) ()
builds the tree with the use of build_Helper()
- void [expression](#) () const
prints out the expressions in the tree with the use of expression_Helper()
- DataType [evaluate](#) () const throw (logic_error)
prints out the total value of the equation stored in the tree with the use of evaluate_Helper()
- void [clear](#) ()
deletes the entirety of the tree with the use of clear_Helper()
- void [commute](#) ()
applies the commutative property with the use of the commute_Helper()
- bool [isEquivalent](#) (const [ExprTree](#) &source) const
- void [showStructure](#) () const

3.1.1 Constructor & Destructor Documentation

3.1.1.1 template<typename DataType > ExprTree< DataType >::ExprTree ()

default constructor for [ExprTree](#) class

Precondition

none

Postcondition

sets root equal to null

Parameters

<i>none</i>	
-------------	--

Returns

none

3.1.1.2 `template<typename DataType > ExprTree< DataType >::ExprTree (const ExprTree< DataType > & source)`

default copy constructor for [ExprTree](#) class

Precondition

copyConstructor_Helper()

Postcondition

sets current tree equal to parameterized tree

Parameters

<i>predefined</i>	ExprTree
-------------------	--------------------------

Returns

*this, which is the current list whether its identical or needs to be rebuilt

3.1.1.3 `template<typename DataType > ExprTree< DataType >::~~ExprTree ()`

default destructor for [ExprTree](#) class

Precondition

[clear\(\)](#)

Postcondition

destroys the current tree

Parameters

<i>none</i>	
-------------	--

Returns

none

3.1.2 Member Function Documentation**3.1.2.1 template<typename DataType > void ExprTree< DataType >::build ()**

builds the tree with the use of build_Helper()

Precondition

build_Helper()

Postcondition

builds an Expression Tree

Parameters

<i>none</i>	
-------------	--

Returns

none

3.1.2.2 template<typename DataType > void ExprTree< DataType >::clear ()

deletes the entirety of the tree with the use of clear_Helper()

Precondition

clear_Helper()

Postcondition

destroys the current tree

Parameters

<i>none</i>	
-------------	--

Returns

none

3.1.2.3 `template<typename DataType > void ExprTree< DataType >::commute ()`

applies the commutative property with the use of the `commute_Helper()`

Precondition

`commute_Helper()`

Postcondition

applies the commutative property to the current expressions stored in the tree

Parameters

<i>none</i>	
-------------	--

Returns

`none`

3.1.2.4 `template<typename DataType > DataType ExprTree< DataType >::evaluate () const throw logic_error`

prints out the total value of the equation stored in the tree with the use of `evaluate_Helper()`

Precondition

`evaluate_Helper()`

Postcondition

`none`

Parameters

<i>none</i>	
-------------	--

Returns

`evaluate_Helper()`, using root as the start location

3.1.2.5 `template<typename DataType > void ExprTree< DataType >::expression () const`

prints out the expressions in the tree with the use of `expression_Helper()`

Precondition

expression_Helper()

Postcondition

none

Parameters

<i>none</i>	
-------------	--

Returns

none

3.1.2.6 `template<typename DataType > bool ExprTree< DataType >::isEquivalent (const ExprTree< DataType > & source) const`

3.1.2.7 `template<typename DataType > ExprTree< DataType > & ExprTree< DataType >::operator= (const ExprTree< DataType > & source)`

overloaded operator for [ExprTree](#) class

Precondition

copyConstructor_Helper()

Postcondition

sets current tree equal to parameterized tree

Parameters

<i>predefined</i>	ExprTree
-------------------	--------------------------

Returns

*this, which is the current list whether its identical or needs to be rebuilt

3.1.2.8 `template<typename DataType > void ExprTree< DataType >::showStructure () const`

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

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

Chapter 4

File Documentation

4.1 ExpressionTree.cpp File Reference

An Implementation file for an Expression Tree.

```
#include "ExpressionTree.h"
```

4.1.1 Detailed Description

An Implementation file for an Expression Tree.

Author

Christopher Eichstedt

4.2 ExpressionTree.h File Reference

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

Classes

- class [ExprTree< DataType >](#)

Index

- ~ExprTree
 - ExprTree, [6](#)
- build
 - ExprTree, [7](#)
- clear
 - ExprTree, [7](#)
- commute
 - ExprTree, [7](#)
- evaluate
 - ExprTree, [8](#)
- ExprTree
 - ~ExprTree, [6](#)
 - build, [7](#)
 - clear, [7](#)
 - commute, [7](#)
 - evaluate, [8](#)
 - ExprTree, [5](#), [6](#)
 - expression, [8](#)
 - isEquivalent, [9](#)
 - operator=, [9](#)
 - showStructure, [9](#)
- ExprTree< DataType >, [5](#)
- expression
 - ExprTree, [8](#)
- ExpressionTree.cpp, [11](#)
- ExpressionTree.h, [11](#)
- isEquivalent
 - ExprTree, [9](#)
- operator=
 - ExprTree, [9](#)
- showStructure
 - ExprTree, [9](#)