TITLE

AUTHOR
Version 1.00
CREATEDATE

Table of Contents

Table of contents

Class Index

Class List

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

BSTClass< DataType > ....pagenum
BSTNode< DataType > ....pagenum
NameType ....pagenum
SimpleTimer ....pagenum

File Index

File List

Here is a list of all files with brief descriptions:

BSTClass.cpp ....pagenum
BSTClass.h (Definition file for Binary Search Tree class) ....pagenum
NameType.cpp (Implementation file for NameType class) ....pagenum
NameType.h (Definition file for NameType class) ....pagenum
PA06.cpp (Driver program to exercise the BST class) ....pagenum
SimpleTimer.cpp (Implementation file for SimpleTimer class) ....pagenum
SimpleTimer.h (Definition file for simple timer class) ....pagenum

Class Documentation

BSTClass< DataType > Class Template Reference

```
#include <BSTClass.h>
```

Public Member Functions

- **BSTClass** ()
- **BSTClass** (const **BSTClass**< DataType > &copied)
- **~BSTClass** ()
- const **BSTClass** & **operator=** (const **BSTClass**< DataType > &rhData)
- void **copyTree** (const **BSTClass**< DataType > &copied)
- void **clearTree** ()
- void **insert** (const DataType &newData)
- bool **findItem** (const DataType &searchDataItem) const
- bool **removeItem** (const DataType &dataItem)
- bool **isEmpty** () const
- void **preOrderTraversal** () const
- void **inOrderTraversal** () const
- void **postOrderTraversal** () const
- int **getHeight** () const
- void **showStructure** () const

Static Public Attributes

- static const char **TAB** = '\t'

Constructor & Destructor Documentation

template<typename DataType > BSTClass< DataType >::BSTClass ()

template<typename DataType > BSTClass< DataType >::BSTClass (const BSTClass< DataType > & *copied*)

template<typename DataType > BSTClass< DataType >::~BSTClass ()

Member Function Documentation

template<typename DataType > void BSTClass< DataType >::clearTree ()

template<typename DataType > void BSTClass< DataType >::copyTree (const BSTClass< DataType > & *copied*)

template<typename DataType > bool BSTClass< DataType >::findItem (const DataType & *searchDataItem*) const

template<typename DataType > int BSTClass< DataType >::getHeight () const

template<typename DataType > void BSTClass< DataType >::inOrderTraversal () const

template<typename DataType > void BSTClass< DataType >::insert (const DataType & *newData*)

template<typename DataType > bool BSTClass< DataType >::isEmpty () const

template<typename DataType > const BSTClass< DataType > & BSTClass< DataType >::operator= (const BSTClass< DataType > & *rhData*)

template<typename DataType > void BSTClass< DataType >::postOrderTraversal () const

template<typename DataType > void BSTClass< DataType >::preOrderTraversal () const

template<typename DataType > bool BSTClass< DataType >::removeItem (const DataType & *dataItem*)

template<typename DataType > void BSTClass< DataType >::showStructure () const

---

Member Data Documentation

template<typename DataType> const char BSTClass< DataType >::TAB = '\t' `[static]`

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

- **BSTClass.h**
- **BSTClass.cpp**

BSTNode< DataType > Class Template Reference

`#include <BSTClass.h>`

Public Member Functions

- **BSTNode** (const DataType &nodeData, **BSTNode** *leftPtr, **BSTNode** *rightPtr)

Public Attributes

- DataType **dataItem**
- **BSTNode**< DataType > * **left**
- **BSTNode**< DataType > * **right**

Constructor & Destructor Documentation

template<typename DataType > BSTNode< DataType >::BSTNode (const DataType & *nodeData*, BSTNode< DataType > * *leftPtr*, BSTNode< DataType > * *rightPtr*)

Member Data Documentation

template<typename DataType> DataType BSTNode< DataType >::dataItem

template<typename DataType> BSTNode<DataType>* BSTNode< DataType >::left

template<typename DataType> BSTNode<DataType>* BSTNode< DataType >::right

---

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

- **BSTClass.h**
- **BSTClass.cpp**

NameType Class Reference

```
#include <NameType.h>
```

Public Member Functions

- **NameType** ()
  Default constructor.
- **NameType** (const char *newName)
  Initialization constructor.
- **NameType** (const **NameType** &newNameObject)
  Copy constructor.
- **~NameType** ()
  Destructor.
- const **NameType** & **operator=** (const **NameType** &rhName)
  Overloaded assignment operator.
- bool **setName** (const char *newName)
  Sets name in data type.
- void **getName** (char *retName) const
  Gets name from data type.
- int **compareTo** (const **NameType** &rhName) const throw (logic_error)
  Compares this name against another.

Static Public Attributes

- static const char **NULL_CHAR** = '\0'
- static const char **COMMA** = ','
- static const char **SPACE** = ' '
- static const int **STD_NAME_LEN** = 100

Constructor & Destructor Documentation

NameType::NameType ()

Default constructor.

Constructs empty **NameType**

Parameters:

None	

Note:

None

NameType::NameType (const char * *newName*)

Initialization constructor.

Places name data into object

Parameters:

in	New string name

Note:

None

NameType::NameType (const NameType & *newNameObject*)

Copy constructor.

Places name data into object

Parameters:

in	New **NameType** object

Note:

None

NameType::~NameType ()

Destructor.

Non-acting destructor, no dynamic data

Parameters:

None	

Note:

None

Member Function Documentation

int NameType::compareTo (const NameType & *rhName*) const throw logic_error)

Compares this name against another.

Return < 0 if this item is less than right hand item Return > 0 if this item is greater than right hand item Return 0 if this item is equal to right hand item

Parameters:

out	returned name

Note:

None

void NameType::getName (char * *retName*) const

Gets name from data type.

Return data as c-string

Parameters:

out	returned name

Note:

None

const NameType & NameType::operator= (const NameType & *rhName*)

Overloaded assignment operator.

Assign data to other **NameType**

Parameters:

in	Assigned name

Note:

None

bool NameType::setName (const char * *newName*)

Sets name in data type.

Assign data to c-string

Parameters:

in	Assigned name

Note:

Attempts to standardize name (LastName, FirstName)

Member Data Documentation

const char NameType::COMMA = ',' `[static]`

const char NameType::NULL_CHAR = '\0' `[static]`

const char NameType::SPACE = ' ' `[static]`

const int NameType::STD_NAME_LEN = 100 `[static]`

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

- **NameType.h**
- **NameType.cpp**

SimpleTimer Class Reference

```
#include <SimpleTimer.h>
```

Public Member Functions

- **SimpleTimer** ()
  Default constructor.
- **~SimpleTimer** ()
  Default constructor.
- void **start** ()
  Start control.
- void **stop** ()
  Stop control.
- void **getElapsedTime** (char *timeStr)

Static Public Attributes

- static const char **NULL_CHAR** = '\0'
- static const char **RADIX_POINT** = '.'

Constructor & Destructor Documentation

SimpleTimer::SimpleTimer ()

Default constructor.

Constructs Timer class

Parameters:

None	

Note:

set running flag to false

SimpleTimer::~SimpleTimer ()

Default constructor.

Destructs Timer class

Parameters:

None	

Note:

No data to clear

Member Function Documentation

void SimpleTimer::getElapsedTime (char * *timeStr*)

void SimpleTimer::start ()

Start control.

Takes initial time data

Parameters:

None	

Note:

None

void SimpleTimer::stop ()

Stop control.

Takes final time data, calculates duration

Parameters:

None	

Note:

None

Member Data Documentation

const char SimpleTimer::NULL_CHAR = '\0' `[static]`

const char SimpleTimer::RADIX_POINT = '.' `[static]`

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

- **SimpleTimer.h**
- **SimpleTimer.cpp**

File Documentation

BSTClass.cpp File Reference

```
#include "BSTClass.h"
#include "NameType.h"
#include <iostream>
```

BSTClass.h File Reference

Definition file for Binary Search Tree class.
```
#include <iostream>
```

Classes

- class **BSTNode< DataType >**
- class **BSTClass< DataType >**

Detailed Description

Definition file for Binary Search Tree class.

Specifies all data of the BST class

Author:

Michael Leverington

Version:

1.00 (03 October 2015)

None

NameType.cpp File Reference

Implementation file for **NameType** class.

```
#include "NameType.h"
#include <iostream>
```

Macros

- #define **CLASS_NAMETYPE_CPP**

Functions

- ostream & **operator<<** (ostream &outStream, const **NameType** &name)
  ostream output operator

Detailed Description

Implementation file for **NameType** class.

Implements the constructor method of the **NameType** class

Author:
Michael Leverington

Version:
1.00 (03 October 2015)

Requires **NameType.h**

Macro Definition Documentation

#define CLASS_NAMETYPE_CPP

Function Documentation

ostream& operator<< (ostream & *outStream*, const NameType & *name*)

ostream output operator

Free function outputs **NameType** to stream

Parameters:

in	ostream file object
in	**NameType** data item

Note:
None

NameType.h File Reference

Definition file for **NameType** class.
```
#include <ostream>
#include <stdexcept>
```

Classes

- class **NameType**

Functions

- ostream & **operator<<** (ostream &outStream, const **NameType** &name)
  ostream output operator

Detailed Description

Definition file for **NameType** class.

Specifies all data of the **NameType** class, along with the constructor, **NameType** class is entered and stored as a string

Author:

Michael Leverington

Version:

1.00 (03 October 2015)

None

Function Documentation

ostream& operator<< (ostream & *outStream*, const NameType & *name*)

ostream output operator

Free function outputs **NameType** to stream

Parameters:

in	ostream file object
in	**NameType** data item

Note:

None

PA06.cpp File Reference

Driver program to exercise the BST class.

```
#include <iostream>
#include <cstring>
#include "NameType.h"
#include "BSTClass.cpp"
```

Functions

- bool **getALine** (istream &consoleIn, char *str)
  Gets name in the form <Last name>="">, <First name>="">
- int **main** ()

Variables

- const char **ENDLINE_CHAR** = '\n'
- const char **CARRIAGE_RETURN_CHAR** = '\r'
- const char **NULL_CHAR** = '\0'
- const int **MAX_NAME_LEN** = 80

Detailed Description

Driver program to exercise the BST class.

Allows for testing the BST class, along with a timer class that will be used for evaluation

Version:

1.00 (3 October 2015)

Requires iostream, cstring, **NameType.h**, and **BSTClass.h**

Function Documentation

bool getALine (istream & *consoleIn*, char * *str*)

Gets name in the form <Last name>="">, <First name>="">

dates are input using cin, and then recombined for string accommodates testing (Submit) system

Parameters:

in	istream object
out	string with date

Note:

resolution for redirected input, getline did not work

int main ()

Variable Documentation

const char CARRIAGE_RETURN_CHAR = '\r'

const char ENDLINE_CHAR = '\n'

const int MAX_NAME_LEN = 80

const char NULL_CHAR = '\0'

SimpleTimer.cpp File Reference

Implementation file for **SimpleTimer** class.
`#include "SimpleTimer.h"`

Macros

- #define **SIMPLETIMER_CPP**

Detailed Description

Implementation file for **SimpleTimer** class.

Author:
Michael Leverington
Implements member methods for timing

Version:
1.00 (11 September 2015)
Requires **SimpleTimer.h**.

Macro Definition Documentation

#define SIMPLETIMER_CPP

SimpleTimer.h File Reference

Definition file for simple timer class.

```
#include <sys/time.h>
#include <cstring>
```

Classes

- class **SimpleTimer**

Detailed Description

Definition file for simple timer class.

Author:

Michael Leverington

Specifies all member methods of the **SimpleTimer**

Version:

1.00 (11 September 2015)

None

Index

INDEX