

My Project

Generated by Doxygen 1.9.6

1 MSDScript	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Class Documentation	9
5.1 AddExpr Class Reference	9
5.1.1 Detailed Description	10
5.1.2 Constructor & Destructor Documentation	10
5.1.2.1 AddExpr()	10
5.1.3 Member Function Documentation	10
5.1.3.1 equals()	10
5.1.3.2 has_variable()	11
5.1.3.3 interp()	11
5.1.3.4 subst()	11
5.1.4 Member Data Documentation	12
5.1.4.1 lhs	12
5.1.4.2 rhs	12
5.2 Expr Class Reference	12
5.2.1 Detailed Description	13
5.2.2 Member Function Documentation	13
5.2.2.1 equals()	13
5.2.2.2 has_variable()	13
5.2.2.3 interp()	13
5.2.2.4 subst()	14
5.3 MultExpr Class Reference	14
5.3.1 Detailed Description	15
5.3.2 Constructor & Destructor Documentation	15
5.3.2.1 MultExpr()	15
5.3.3 Member Function Documentation	15
5.3.3.1 equals()	15
5.3.3.2 has_variable()	17
5.3.3.3 interp()	17
5.3.3.4 subst()	17
5.3.4 Member Data Documentation	18
5.3.4.1 lhs	18
5.3.4.2 rhs	18
5.4 NumExpr Class Reference	18

5.4.1 Detailed Description	19
5.4.2 Constructor & Destructor Documentation	19
5.4.2.1 NumExpr()	19
5.4.3 Member Function Documentation	20
5.4.3.1 equals()	20
5.4.3.2 has_variable()	20
5.4.3.3 interp()	20
5.4.3.4 subst()	20
5.4.4 Member Data Documentation	21
5.4.4.1 val	21
5.5 VarExpr Class Reference	21
5.5.1 Detailed Description	22
5.5.2 Constructor & Destructor Documentation	22
5.5.2.1 VarExpr()	22
5.5.3 Member Function Documentation	23
5.5.3.1 equals()	23
5.5.3.2 has_variable()	23
5.5.3.3 interp()	23
5.5.3.4 subst()	23
5.5.4 Member Data Documentation	24
5.5.4.1 val	24
6 File Documentation	25
6.1 /Users/randiprince/cs6015/MSDscript/MSDscript/commandLine.cpp File Reference	25
6.1.1 Detailed Description	25
6.1.2 Function Documentation	25
6.1.2.1 use_arguments()	25
6.2 /Users/randiprince/cs6015/MSDscript/MSDscript/commandLine.hpp File Reference	26
6.2.1 Detailed Description	26
6.2.2 Function Documentation	26
6.2.2.1 use_arguments()	26
6.3 /Users/randiprince/cs6015/MSDscript/MSDscript/commandLine.hpp	26
6.4 /Users/randiprince/cs6015/MSDscript/MSDscript/expr.cpp File Reference	27
6.4.1 Detailed Description	27
6.5 /Users/randiprince/cs6015/MSDscript/MSDscript/expr.hpp File Reference	27
6.5.1 Detailed Description	27
6.6 /Users/randiprince/cs6015/MSDscript/MSDscript/expr.hpp	28
6.7 /Users/randiprince/cs6015/MSDscript/MSDscript/tests.cpp File Reference	28
6.7.1 Detailed Description	29
6.7.2 Function Documentation	29
6.7.2.1 TEST_CASE() [1/7]	29
6.7.2.2 TEST_CASE() [2/7]	29

6.7.2.3 TEST_CASE() [3/7]	29
6.7.2.4 TEST_CASE() [4/7]	29
6.7.2.5 TEST_CASE() [5/7]	30
6.7.2.6 TEST_CASE() [6/7]	30
6.7.2.7 TEST_CASE() [7/7]	30
Index	31

Chapter 1

MSDScript

Author

Randi Prince

Date

02-02-2023

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Expr	12
AddExpr	9
MultExpr	14
NumExpr	18
VarExpr	21

Chapter 3

Class Index

3.1 Class List

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

AddExpr	9
Expr	12
MultExpr	14
NumExpr	18
VarExpr	21

Chapter 4

File Index

4.1 File List

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

/Users/randiprince/cs6015/MSDscript/MSDscript/commandLine.cpp	
Use_argument function definition	25
/Users/randiprince/cs6015/MSDscript/MSDscript/commandLine.hpp	
Use_argument function declaration	26
/Users/randiprince/cs6015/MSDscript/MSDscript/expr.cpp	
Expression class definitions for all the sub classes of Expr class	27
/Users/randiprince/cs6015/MSDscript/MSDscript/expr.hpp	
Expression class declarations for all the sub classes of Expr class	27
/Users/randiprince/cs6015/MSDscript/MSDscript/tests.cpp	
Tests for msdscript program	28

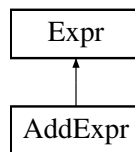
Chapter 5

Class Documentation

5.1 AddExpr Class Reference

```
#include <expr.hpp>
```

Inheritance diagram for AddExpr:



Public Member Functions

- `AddExpr (Expr *lhs, Expr *rhs)`
- `bool equals (Expr *expr)`
determines if one Expr is equal to another Expr
- `int interp ()`
returns an int for the value of an expression.
- `bool has_variable ()`
function to determine if the expr has a varExpr
- `Expr * subst (std::string s, Expr *e)`
Everywhere that the expression (whose subst method is called) contains a variable matching the string, the result Expr should have the given replacement, instead.*
- `virtual bool equals (Expr *e)=0`
determines if one Expr is equal to another Expr
- `virtual int interp ()=0`
returns an int for the value of an expression.
- `virtual bool has_variable ()=0`
function to determine if the expr has a varExpr
- `virtual Expr * subst (std::string s, Expr *e)=0`
Everywhere that the expression (whose subst method is called) contains a variable matching the string, the result Expr should have the given replacement, instead.*

Public Attributes

- [Expr](#) * lhs
- [Expr](#) * rhs

5.1.1 Detailed Description

@brief [AddExpr](#) class This is a sub class that implements [Expr](#). It adds two expr together It implements all virtual methods of [Expr](#).

5.1.2 Constructor & Destructor Documentation

5.1.2.1 AddExpr()

```
AddExpr::AddExpr (
    Expr * lhs,
    Expr * rhs )
```

Constructor that creates new [AddExpr](#).

Parameters

<i>lhs</i>	a pointer to an Expr
<i>rhs</i>	a pointer to an Expr

5.1.3 Member Function Documentation

5.1.3.1 equals()

```
bool AddExpr::equals (
    Expr * e ) [virtual]
```

determines if one [Expr](#) is equal to another [Expr](#)

Parameters

<i>e</i>	to compare the value of
----------	-------------------------

Returns

boolean

Implements [Expr](#).

5.1.3.2 has_variable()

```
bool AddExpr::has_variable ( ) [virtual]
```

function to determine if the expr has a varExpr

Returns

boolean

Implements [Expr](#).

5.1.3.3 interp()

```
int AddExpr::interp ( ) [virtual]
```

returns an int for the value of an expression.

Returns

int

Implements [Expr](#).

5.1.3.4 subst()

```
Expr * AddExpr::subst (
    std::string s,
    Expr * e ) [virtual]
```

Everywhere that the expression (whose subst method is called) contains a variable matching the string, the result Expr* should have the given replacement, instead.

Parameters

<i>s</i>	
<i>e</i>	pointer

Returns

[Expr](#)

Implements [Expr](#).

5.1.4 Member Data Documentation

5.1.4.1 lhs

[Expr](#)* [AddExpr::lhs](#)

left hand side expr of [AddExpr](#)

5.1.4.2 rhs

[Expr](#)* [AddExpr::rhs](#)

right hand side expr of [AddExpr](#)

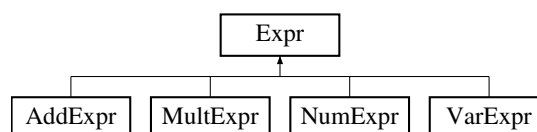
The documentation for this class was generated from the following files:

- [/Users/randiprince/cs6015/MSDscript/MSDscript/expr.hpp](#)
- [/Users/randiprince/cs6015/MSDscript/MSDscript/expr.cpp](#)

5.2 Expr Class Reference

```
#include <expr.hpp>
```

Inheritance diagram for Expr:



Public Member Functions

- virtual bool [equals](#) ([Expr](#) *e)=0
determines if one [Expr](#) is equal to another [Expr](#)
- virtual int [interp](#) ()=0
returns an int for the value of an expression.
- virtual bool [has_variable](#) ()=0
function to determine if the expr has a varExpr
- virtual [Expr](#) * [subst](#) (std::string s, [Expr](#) *e)=0
Everywhere that the expression (whose subst method is called) contains a variable matching the string, the result Expr should have the given replacement, instead.*

5.2.1 Detailed Description

@brief [Expr](#) class This is a parent class called [Expr](#). It has virtual methods used by sub classes.

5.2.2 Member Function Documentation

5.2.2.1 equals()

```
virtual bool Expr::equals (
    Expr * e ) [pure virtual]
```

determines if one [Expr](#) is equal to another [Expr](#)

Parameters

e	to compare the value of
-------------------	-------------------------

Returns

boolean

Implemented in [NumExpr](#), [AddExpr](#), [MultExpr](#), and [VarExpr](#).

5.2.2.2 has_variable()

```
virtual bool Expr::has_variable ( ) [pure virtual]
```

function to determine if the expr has a varExpr

Returns

boolean

Implemented in [NumExpr](#), [AddExpr](#), [MultExpr](#), and [VarExpr](#).

5.2.2.3 interp()

```
virtual int Expr::interp ( ) [pure virtual]
```

returns an int for the value of an expression.

Returns

int

Implemented in [NumExpr](#), [AddExpr](#), [MultExpr](#), and [VarExpr](#).

5.2.2.4 subst()

```
virtual Expr * Expr::subst (
    std::string s,
    Expr * e ) [pure virtual]
```

Everywhere that the expression (whose subst method is called) contains a variable matching the string, the result Expr* should have the given replacement, instead.

Parameters

<i>s</i>	
<i>e</i>	pointer

Returns

Expr

Implemented in [NumExpr](#), [AddExpr](#), [MultExpr](#), and [VarExpr](#).

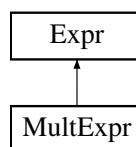
The documentation for this class was generated from the following file:

- [/Users/randiprince/cs6015/MSDscript/MSDscript/expr.hpp](#)

5.3 MultExpr Class Reference

```
#include <expr.hpp>
```

Inheritance diagram for MultExpr:



Public Member Functions

- [MultExpr](#) (Expr *lhs, Expr *rhs)
- bool [equals](#) (Expr *expr)
determines if one Expr is equal to another Expr
- int [interp](#) ()
returns an int for the value of an expression.
- bool [has_variable](#) ()
function to determine if the expr has a varExpr
- Expr * [subst](#) (std::string s, Expr *e)
Everywhere that the expression (whose subst method is called) contains a variable matching the string, the result Expr should have the given replacement, instead.*

- virtual bool `equals (Expr *e)=0`
determines if one Expr is equal to another Expr
- virtual int `interp ()=0`
returns an int for the value of an expression.
- virtual bool `has_variable ()=0`
function to determine if the expr has a varExpr
- virtual `Expr * subst (std::string s, Expr *e)=0`
Everywhere that the expression (whose subst method is called) contains a variable matching the string, the result Expr should have the given replacement, instead.*

Public Attributes

- `Expr * lhs`
- `Expr * rhs`

5.3.1 Detailed Description

@brief `MultExpr` class This is a sub class that implements `Expr`. It multiplies two expr together It implements all virtual methods of `expr`.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 MultExpr()

```
MultExpr::MultExpr (
    Expr * lhs,
    Expr * rhs )
```

Constructor that creates new `MultExpr`.

Parameters

<i>lhs</i>	a pointer to an <code>Expr</code>
<i>rhs</i>	a pointer to an <code>Expr</code>

5.3.3 Member Function Documentation

5.3.3.1 equals()

```
bool MultExpr::equals (
    Expr * e ) [virtual]
```

determines if one [Expr](#) is equal to another [Expr](#)

Parameters

<code>e</code>	to compare the value of
----------------	-------------------------

Returns

boolean

Implements [Expr](#).

5.3.3.2 has_variable()

```
bool MultExpr::has_variable ( ) [virtual]
```

function to determine if the expr has a varExpr

Returns

boolean

Implements [Expr](#).

5.3.3.3 interp()

```
int MultExpr::interp ( ) [virtual]
```

returns an int for the value of an expression.

Returns

int

Implements [Expr](#).

5.3.3.4 subst()

```
Expr * MultExpr::subst (
    std::string s,
    Expr * e ) [virtual]
```

Everywhere that the expression (whose subst method is called) contains a variable matching the string, the result Expr* should have the given replacement, instead.

Parameters

s	
e	pointer

Returns

[Expr](#)

Implements [Expr](#).

5.3.4 Member Data Documentation

5.3.4.1 lhs

```
Expr* MultExpr::lhs
```

left hand side expr of [MultExpr](#)

5.3.4.2 rhs

```
Expr* MultExpr::rhs
```

right hand side expr of [MultExpr](#)

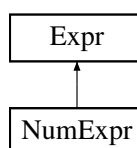
The documentation for this class was generated from the following files:

- [/Users/randiprince/cs6015/MSDscript/MSDscript/expr.hpp](#)
- [/Users/randiprince/cs6015/MSDscript/MSDscript/expr.cpp](#)

5.4 NumExpr Class Reference

```
#include <expr.hpp>
```

Inheritance diagram for NumExpr:



Public Member Functions

- [NumExpr](#) (int [val](#))
constructor of [NumExpr](#) class
- bool [equals](#) ([Expr](#) *[expr](#))
determines if one [Expr](#) is equal to another [Expr](#)
- int [interp](#) ()
returns an int for the value of an expression.
- bool [has_variable](#) ()
function to determine if the expr has a varExpr
- [Expr](#) * [subst](#) (std::string [s](#), [Expr](#) *[e](#))
Everywhere that the expression (whose subst method is called) contains a variable matching the string, the result Expr should have the given replacement, instead.*
- virtual bool [equals](#) ([Expr](#) *[e](#))=0
determines if one [Expr](#) is equal to another [Expr](#)
- virtual int [interp](#) ()=0
returns an int for the value of an expression.
- virtual bool [has_variable](#) ()=0
function to determine if the expr has a varExpr
- virtual [Expr](#) * [subst](#) (std::string [s](#), [Expr](#) *[e](#))=0
Everywhere that the expression (whose subst method is called) contains a variable matching the string, the result Expr should have the given replacement, instead.*

Public Attributes

- int [val](#)

5.4.1 Detailed Description

@brief [NumExpr](#) class This is a sub class that implements [Expr](#). It is an expr that is an integer It implements all virtual methods of expr.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 NumExpr()

```
NumExpr::NumExpr (
    int val )
```

constructor of [NumExpr](#) class

Parameters

val	
---------------------	--

5.4.3 Member Function Documentation

5.4.3.1 equals()

```
bool NumExpr::equals (
    Expr * e ) [virtual]
```

determines if one [Expr](#) is equal to another [Expr](#)

Parameters

<i>e</i>	to compare the value of
----------	-------------------------

Returns

boolean

Implements [Expr](#).

5.4.3.2 has_variable()

```
bool NumExpr::has_variable ( ) [virtual]
```

function to determine if the expr has a varExpr

Returns

boolean

Implements [Expr](#).

5.4.3.3 interp()

```
int NumExpr::interp ( ) [virtual]
```

returns an int for the value of an expression.

Returns

int

Implements [Expr](#).

5.4.3.4 subst()

```
Expr * NumExpr::subst (
    std::string s,
    Expr * e ) [virtual]
```

Everywhere that the expression (whose subst method is called) contains a variable matching the string, the result Expr* should have the given replacement, instead.

Parameters

s	
e	pointer

Returns

[Expr](#)

Implements [Expr](#).

5.4.4 Member Data Documentation

5.4.4.1 val

```
int NumExpr::val
```

integer value member variable of [NumExpr](#) class

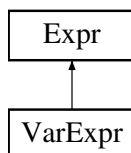
The documentation for this class was generated from the following files:

- [/Users/randiprince/cs6015/MSDscript/MSDscript/expr.hpp](#)
- [/Users/randiprince/cs6015/MSDscript/MSDscript/expr.cpp](#)

5.5 VarExpr Class Reference

```
#include <expr.hpp>
```

Inheritance diagram for VarExpr:



Public Member Functions

- [VarExpr](#) (std::string *val*)
- bool [equals](#) ([Expr](#) **expr*)
determines if one [Expr](#) is equal to another [Expr](#)
- int [interp](#) ()
returns an int for the value of an expression.
- bool [has_variable](#) ()
function to determine if the expr has a [VarExpr](#)
- [Expr](#) * [subst](#) (std::string *s*, [Expr](#) **e*)
Everywhere that the expression (whose subst method is called) contains a variable matching the string, the result [Expr](#) should have the given replacement, instead.*

- virtual bool [equals](#) ([Expr](#) **e*)=0
determines if one [Expr](#) is equal to another [Expr](#)
- virtual int [interp](#) ()=0
returns an int for the value of an expression.
- virtual bool [has_variable](#) ()=0
function to determine if the expr has a [VarExpr](#)
- virtual [Expr](#) * [subst](#) (std::string *s*, [Expr](#) **e*)=0
Everywhere that the expression (whose subst method is called) contains a variable matching the string, the result [Expr](#) should have the given replacement, instead.*

Public Attributes

- std::string [val](#)

5.5.1 Detailed Description

@brief [VarExpr](#) class This is a sub class that implements [Expr](#). It is an expr that is a string. It implements all virtual methods of [Expr](#).

5.5.2 Constructor & Destructor Documentation

5.5.2.1 [VarExpr](#)()

```
VarExpr::VarExpr (
    std::string val )
```

Constructor that creates new [VarExpr](#).

Parameters

<i>val</i>	is a string
------------	-------------

5.5.3 Member Function Documentation

5.5.3.1 equals()

```
bool VarExpr::equals (
    Expr * e ) [virtual]
```

determines if one [Expr](#) is equal to another [Expr](#)

Parameters

<i>e</i>	to compare the value of
----------	-------------------------

Returns

boolean

Implements [Expr](#).

5.5.3.2 has_variable()

```
bool VarExpr::has_variable ( ) [virtual]
```

function to determine if the expr has a varExpr

Returns

boolean

Implements [Expr](#).

5.5.3.3 interp()

```
int VarExpr::interp ( ) [virtual]
```

returns an int for the value of an expression.

Returns

int

Implements [Expr](#).

5.5.3.4 subst()

```
Expr * VarExpr::subst (
    std::string s,
    Expr * e ) [virtual]
```

Everywhere that the expression (whose subst method is called) contains a variable matching the string, the result Expr* should have the given replacement, instead.

Parameters

<i>s</i>	
<i>e</i>	pointer

Returns

[Expr](#)

Implements [Expr](#).

5.5.4 Member Data Documentation

5.5.4.1 `val`

```
std::string VarExpr::val
```

String called val, which is a member variable of [VarExpr](#)

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

- [/Users/randiprince/cs6015/MSDscript/MSDscript/expr.hpp](#)
- [/Users/randiprince/cs6015/MSDscript/MSDscript/expr.cpp](#)

Chapter 6

File Documentation

6.1 /Users/randiprince/cs6015/MSDscript/MSDscript/commandLine.cpp File Reference

contains use_argument function definition

```
#include "commandLine.hpp"
#include "catch.h"
#include <iostream>
#include <cstdlib>
```

Functions

- void [use_arguments](#) (int argc, char *argv[])
function to call in main

6.1.1 Detailed Description

contains use_argument function definition

Author

Randi Prince

6.1.2 Function Documentation

6.1.2.1 use_arguments()

```
void use_arguments (
    int argc,
    char * argv[] )
```

function to call in main

Parameters

<i>argc</i>	
<i>argv</i>	

6.2 /Users/randiprince/cs6015/MSDscript/MSDscript/commandLine.hpp File Reference

contains use_argument function declaration

Functions

- void [use_arguments](#) (int argc, char *argv[])
function to call in main

6.2.1 Detailed Description

contains use_argument function declaration

Author

Randi Prince

6.2.2 Function Documentation

6.2.2.1 use_arguments()

```
void use_arguments (
    int argc,
    char * argv[] )
```

function to call in main

Parameters

<i>argc</i>	
<i>argv</i>	

6.3 /Users/randiprince/cs6015/MSDscript/MSDscript/commandLine.hpp

[Go to the documentation of this file.](#)


```
00001 //
00002 //  commandLine.hpp
00003 //  MSDscript
00004 //
00005 //  Created by Randi Prince on 1/15/23.
00006 //
00013 #ifndef commandLine_hpp
00014 #define commandLine_hpp
00015
00016
00022 void use_arguments(int argc, char * argv[]);
00023
00024 #endif /* commandLine_hpp */
```

6.4 /Users/randiprince/cs6015/MSDscript/MSDscript/expr.cpp File Reference

contains expression class definitions for all the sub classes of [Expr](#) class

```
#include "expr.hpp"
#include <stdexcept>
```

6.4.1 Detailed Description

contains expression class definitions for all the sub classes of [Expr](#) class

Author

Randi Prince

6.5 /Users/randiprince/cs6015/MSDscript/MSDscript/expr.hpp File Reference

contains expression class declarations for all the sub classes of [Expr](#) class

```
#include <string>
```

Classes

- class [Expr](#)
- class [NumExpr](#)
- class [AddExpr](#)
- class [MultExpr](#)
- class [VarExpr](#)

6.5.1 Detailed Description

contains expression class declarations for all the sub classes of [Expr](#) class

Author

Randi Prince

6.6 /Users/randiprince/cs6015/MSDscript/MSDscript/expr.hpp

[Go to the documentation of this file.](#)

```

00001 //
00002 //  expr.hpp
00003 //  MSDscript
00004 //
00005 //  Created by Randi Prince on 1/23/23.
00006 //
00007
00014 #ifndef expr_hpp
00015 #define expr_hpp
00016
00017 #include <string>
00018
00024 class Expr {
00025 public:
00031     virtual bool equals(Expr *e) = 0;
00032
00037     virtual int interp() = 0;
00042     virtual bool has_variable() = 0;
00043
00051     virtual Expr* subst(std::string s, Expr *e) = 0;
00052 };
00053
00059 class NumExpr : public Expr {
00060 public:
00061     int val;
00067     NumExpr(int val);
00068     bool equals(Expr *expr);
00069     int interp();
00070     bool has_variable();
00071     Expr* subst(std::string s, Expr *e);
00072 };
00073
00079 class AddExpr : public Expr {
00080 public:
00081     Expr *lhs;
00082     Expr *rhs;
00089     AddExpr(Expr *lhs, Expr *rhs);
00090     bool equals(Expr *expr);
00091     int interp();
00092     bool has_variable();
00093     Expr* subst(std::string s, Expr *e);
00094 };
00095
00101 class MultExpr : public Expr {
00102 public:
00103     Expr *lhs;
00104     Expr *rhs;
00110     MultExpr(Expr *lhs, Expr *rhs);
00111     bool equals(Expr *expr);
00112     int interp();
00113     bool has_variable();
00114     Expr* subst(std::string s, Expr *e);
00115 };
00116
00122 class VarExpr : public Expr {
00123 public:
00124     std::string val;
00130     VarExpr(std::string val);
00131
00132     bool equals(Expr *expr);
00133     int interp();
00134     bool has_variable();
00135     Expr* subst(std::string s, Expr *e);
00136 };
00137
00138 #endif /* expr_hpp */

```

6.7 /Users/randiprince/cs6015/MSDscript/MSDscript/tests.cpp File Reference

contains tests for msdscript program

```

#include "catch.h"
#include "expr.hpp"

```

Functions

- `TEST_CASE` ("Test `NumExpr` equals")
- `TEST_CASE` ("Test `AddExpr` equals")
- `TEST_CASE` ("Test `MultExpr` equals")
- `TEST_CASE` ("Test `VarExpr` equals")
- `TEST_CASE` ("Interp Function")
- `TEST_CASE` ("Has `VarExpr` Function")
- `TEST_CASE` ("Substitution (subst) function")

6.7.1 Detailed Description

contains tests for msdscript program

Author

Randi Prince

6.7.2 Function Documentation

6.7.2.1 `TEST_CASE()` [1/7]

```
TEST_CASE (
    "Has VarExpr Function" )
```

@brief Test has_variable function Tests of has_variable function for each class

6.7.2.2 `TEST_CASE()` [2/7]

```
TEST_CASE (
    "Interp Function" )
```

@brief Test Interp function Tests of interp function for each class

6.7.2.3 `TEST_CASE()` [3/7]

```
TEST_CASE (
    "Substitution (subst) function" )
```

@brief Test subst function Tests of subst function for each class

6.7.2.4 `TEST_CASE()` [4/7]

```
TEST_CASE (
    "Test AddExpr equals" )
```

@brief Test `AddExpr` equals Tests of `AddExpr` equals function

6.7.2.5 TEST_CASE() [5/7]

```
TEST_CASE (
    "Test MultExpr equals" )
```

@brief Test MultExpr equals Tests of MultExpr equals function

6.7.2.6 TEST_CASE() [6/7]

```
TEST_CASE (
    "Test NumExpr equals" )
```

@brief Test NumExpr equals Tests of numExpr equals function

6.7.2.7 TEST_CASE() [7/7]

```
TEST_CASE (
    "Test VarExpr equals" )
```

@brief Test VarExpr equals Tests of VarExpr equals function

Index

`/Users/randiprince/cs6015/MSDscript/MSDscript/commandLine.cpp`,
25
`/Users/randiprince/cs6015/MSDscript/MSDscript/commandLine.hpp`,
26
`/Users/randiprince/cs6015/MSDscript/MSDscript/expr.cpp`,
27
`/Users/randiprince/cs6015/MSDscript/MSDscript/expr.hpp`,
27
`/Users/randiprince/cs6015/MSDscript/MSDscript/tests.cpp`,
28

`AddExpr`, 9
 `AddExpr`, 10
 `equals`, 10
 `has_variable`, 11
 `interp`, 11
 `lhs`, 12
 `rhs`, 12
 `subst`, 11

`commandLine.cpp`
 `use_arguments`, 25
`commandLine.hpp`
 `use_arguments`, 26

`equals`
 `AddExpr`, 10
 `Expr`, 13
 `MultExpr`, 15
 `NumExpr`, 20
 `VarExpr`, 23

`Expr`, 12
 `equals`, 13
 `has_variable`, 13
 `interp`, 13
 `subst`, 13

`has_variable`
 `AddExpr`, 11
 `Expr`, 13
 `MultExpr`, 17
 `NumExpr`, 20
 `VarExpr`, 23

`interp`
 `AddExpr`, 11
 `Expr`, 13
 `MultExpr`, 17
 `NumExpr`, 20
 `VarExpr`, 23

`MultExpr`, 14
 `equals`, 15
 `has_variable`, 17
 `interp`, 17
 `lhs`, 18
 `MultExpr`, 15
 `rhs`, 18
 `subst`, 17

`NumExpr`, 18
 `equals`, 20
 `has_variable`, 20
 `interp`, 20
 `NumExpr`, 19
 `subst`, 20
 `val`, 21

`rhs`
 `AddExpr`, 12
 `MultExpr`, 18

`subst`
 `AddExpr`, 11
 `Expr`, 13
 `MultExpr`, 17
 `NumExpr`, 20
 `VarExpr`, 23

`TEST_CASE`
 `tests.cpp`, 29, 30

`tests.cpp`
 `TEST_CASE`, 29, 30

`use_arguments`
 `commandLine.cpp`, 25
 `commandLine.hpp`, 26

`val`
 `NumExpr`, 21
 `VarExpr`, 24

`VarExpr`, 21
 `equals`, 23
 `has_variable`, 23
 `interp`, 23
 `subst`, 23
 `val`, 24
 `VarExpr`, 22