

MSDScript

Generated by Doxygen 1.9.6



<b>1 MSDScript</b>	<b>1</b>
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List	5
<b>4 File Index</b>	<b>7</b>
4.1 File List	7
<b>5 Class Documentation</b>	<b>9</b>
5.1 Add Class Reference	9
5.1.1 Constructor & Destructor Documentation	9
5.1.1.1 Add()	9
5.1.2 Member Function Documentation	10
5.1.2.1 equals()	10
5.1.2.2 has_variable()	10
5.1.2.3 interp()	11
5.1.2.4 subst()	11
5.2 Expr Class Reference	11
5.2.1 Member Function Documentation	12
5.2.1.1 equals()	12
5.2.1.2 has_variable()	12
5.2.1.3 interp()	12
5.2.1.4 subst()	12
5.3 Mult Class Reference	13
5.3.1 Constructor & Destructor Documentation	13
5.3.1.1 Mult()	13
5.3.2 Member Function Documentation	14
5.3.2.1 equals()	14
5.3.2.2 has_variable()	14
5.3.2.3 interp()	14
5.3.2.4 subst()	14
5.4 Num Class Reference	15
5.4.1 Constructor & Destructor Documentation	15
5.4.1.1 Num()	15
5.4.2 Member Function Documentation	16
5.4.2.1 equals()	16
5.4.2.2 has_variable()	16
5.4.2.3 interp()	16
5.4.2.4 subst()	17
5.5 Var Class Reference	17
5.5.1 Constructor & Destructor Documentation	18

---

5.5.1.1 Var()	18
5.5.2 Member Function Documentation	18
5.5.2.1 equals()	18
5.5.2.2 has_variable()	18
5.5.2.3 interp()	19
5.5.2.4 subst()	19
<b>6 File Documentation</b>	<b>21</b>
6.1 /Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/cmdline.h File Reference	21
6.1.1 Detailed Description	21
6.2 /Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/cmdline.h . . .	21
6.3 /Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.cpp File Reference	22
6.3.1 Detailed Description	22
6.4 /Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.h File Reference	22
6.4.1 Detailed Description	22
6.5 /Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.h . . . .	22
<b>Index</b>	<b>25</b>

# Chapter 1

## MSDScript

Author

Jowie Tan

Date

02-07-2023



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

Expr . . . . .	11
Add . . . . .	9
Mult . . . . .	13
Num . . . . .	15
Var . . . . .	17





## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">Add</a>	.....	9
<a href="#">Expr</a>	.....	11
<a href="#">Mult</a>	.....	13
<a href="#">Num</a>	.....	15
<a href="#">Var</a>	.....	17



## Chapter 4

# File Index

### 4.1 File List

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

/Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/cmdline.h	
Contain functions to run program from terminal . . . . .	21
/Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.cpp	
Contains all the expression methods that are inherited by <a href="#">Num</a> , <a href="#">Add</a> , <a href="#">Mult</a> and <a href="#">Var</a> . . . . .	22
/Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.h	
Contains the methods that would be defined in <a href="#">expr.cpp</a> . . . . .	22



## Chapter 5

# Class Documentation

### 5.1 Add Class Reference

Inheritance diagram for Add:



#### Public Member Functions

- `Add (Expr *lhs, Expr *rhs)`  
*Sets value to add expression.*
- `bool equals (Expr *other)`  
*Compare Value.*
- `int interp ()`  
*Interprets the Add expression.*
- `bool has_variable ()`  
*check if expression has variable*
- `Expr * subst (std::string target, Expr *other)`  
*if variable matches target string, replaces variable with Expr other. This goes into each expression in each side*
- `virtual bool equals (Expr *e)=0`
- `virtual int interp ()=0`
- `virtual bool has_variable ()=0`
- `virtual Expr * subst (std::string target, Expr *other)=0`

#### 5.1.1 Constructor & Destructor Documentation

##### 5.1.1.1 Add()

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

Sets value to add expression.

**Parameters**

<i>Expr</i>	left value
<i>Expr</i>	right value

## 5.1.2 Member Function Documentation

### 5.1.2.1 equals()

```
bool Add::equals (
    Expr * other ) [virtual]
```

Compare Value.

**Parameters**

<i>Expr</i>	other
-------------	-------

**Returns**

Boolean result

Implements *Expr*.

### 5.1.2.2 has\_variable()

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

check if expression has variable

**Returns**

Boolean results

Implements *Expr*.

### 5.1.2.3 interp()

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

Interprets the [Add](#) expression.

#### Returns

Int add the value of the left expression and right expression

Implements [Expr](#).

### 5.1.2.4 subst()

```
Expr * Add::subst (
    std::string target,
    Expr * other ) [virtual]
```

if variable matches target string, replaces variable with [Expr](#) other. This goes into each expression in each side

#### Parameters

<i>string</i>	target - target string to replace
<i>Expr*</i>	other - replacement expression

#### Returns

new [Add](#) expression

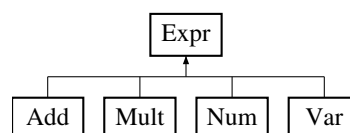
Implements [Expr](#).

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

- </Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.h>
- </Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.cpp>

## 5.2 Expr Class Reference

Inheritance diagram for Expr:



## Public Member Functions

- virtual bool [equals](#) ([Expr](#) \*e)=0
- virtual int [interp](#) ()=0
- virtual bool [has\\_variable](#) ()=0
- virtual [Expr](#) \* [subst](#) (std::string target, [Expr](#) \*other)=0

### 5.2.1 Member Function Documentation

#### 5.2.1.1 equals()

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

Implemented in [Num](#), [Add](#), [Mult](#), and [Var](#).

#### 5.2.1.2 has\_variable()

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

Implemented in [Num](#), [Add](#), [Mult](#), and [Var](#).

#### 5.2.1.3 interp()

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

Implemented in [Num](#), [Add](#), [Mult](#), and [Var](#).

#### 5.2.1.4 subst()

```
virtual Expr * Expr::subst (  
    std::string target,  
    Expr * other ) [pure virtual]
```

Implemented in [Num](#), [Add](#), [Mult](#), and [Var](#).

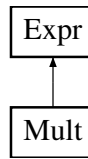
The documentation for this class was generated from the following file:

- [/Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.h](#)



## 5.3 Mult Class Reference

Inheritance diagram for Mult:



### Public Member Functions

- `Mult (Expr *lhs, Expr *rhs)`  
*Sets value to mult expression.*
- `bool equals (Expr *other)`  
*Compare Value.*
- `int interp ()`  
*Interprets the Mult expression.*
- `bool has_variable ()`  
*check if expression has variable*
- `Expr * subst (std::string target, Expr *other)`  
*if variable matches target string, replaces variable with Expr other. This goes into each expression in each side*
- `virtual bool equals (Expr *e)=0`
- `virtual int interp ()=0`
- `virtual bool has_variable ()=0`
- `virtual Expr * subst (std::string target, Expr *other)=0`

### 5.3.1 Constructor & Destructor Documentation

#### 5.3.1.1 Mult()

```

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

Sets value to mult expression.

#### Parameters

<code>Expr*</code>	left value
<code>Expr*</code>	right value

## 5.3.2 Member Function Documentation

### 5.3.2.1 equals()

```
bool Mult::equals (
    Expr * other ) [virtual]
```

Compare Value.

#### Parameters

<i>Expr*</i>	other
--------------	-------

#### Returns

Boolean result

Implements [Expr](#).

### 5.3.2.2 has\_variable()

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

check if expression has variable

#### Returns

Boolean results

Implements [Expr](#).

### 5.3.2.3 interp()

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

Interprets the [Mult](#) expression.

#### Returns

Int times the value of the left expression and right expression

Implements [Expr](#).

### 5.3.2.4 subst()

```
Expr * Mult::subst (
    std::string target,
    Expr * other ) [virtual]
```

if variable matches target string, replaces variable with [Expr](#) other. This goes into each expression in each side

## Parameters

<i>string</i>	target - target string to replace
<i>Expr*</i>	other - replacement expression

## Returns

new [Mult](#) expression

Implements [Expr](#).

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

- [/Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.h](#)
- [/Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.cpp](#)

## 5.4 Num Class Reference

Inheritance diagram for Num:



### Public Member Functions

- [Num](#) (int val)  
*Sets value.*
- bool [equals](#) ([Expr](#) \*other)  
*Compare Value.*
- int [interp](#) ()  
*Interprets the [Num](#) expression.*
- bool [has\\_variable](#) ()  
*check if expression has variable*
- [Expr](#) \* [subst](#) (std::string target, [Expr](#) \*other)  
*if variable matches target string, replaces variable with [Expr](#) other. However, since this is not a variable, there is no change.*
- virtual bool [equals](#) ([Expr](#) \*e)=0
- virtual int [interp](#) ()=0
- virtual bool [has\\_variable](#) ()=0
- virtual [Expr](#) \* [subst](#) (std::string target, [Expr](#) \*other)=0

### 5.4.1 Constructor & Destructor Documentation

#### 5.4.1.1 Num()

```
Num::Num (
    int val )
```

Sets value.

**Parameters**

<i>value</i>	
--------------	--

## 5.4.2 Member Function Documentation

### 5.4.2.1 equals()

```
bool Num::equals (
    Expr * other ) [virtual]
```

Compare Value.

**Parameters**

<i>Expr</i>	other
-------------	-------

**Returns**

Boolean result

Implements [Expr](#).

### 5.4.2.2 has\_variable()

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

check if expression has variable

**Returns**

Boolean false

Implements [Expr](#).

### 5.4.2.3 interp()

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

Interprets the [Num](#) expression.

**Returns**

int Value

Implements [Expr](#).

#### 5.4.2.4 subst()

```
Expr * Num::subst (
    std::string target,
    Expr * other ) [virtual]
```

if variable matches target string, replaces variable with [Expr](#) other. However, since this is not a variable, there is no change.

##### Parameters

<i>string</i>	target - target string to replace
<i>Expr*</i>	other - replacement expression

##### Returns

this (no change)

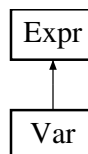
Implements [Expr](#).

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

- [/Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.h](#)
- [/Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.cpp](#)

## 5.5 Var Class Reference

Inheritance diagram for Var:



### Public Member Functions

- [Var](#) (std::string value)  
*Sets String.*
- bool [equals](#) ([Expr](#) \*other)  
*Compare Value.*
- int [interp](#) ()  
*Interprets the variable expression.*
- bool [has\\_variable](#) ()  
*check if expression has variable*
- [Expr](#) \* [subst](#) (std::string target, [Expr](#) \*other)  
*if variable matches target string, replaces variable with [Expr](#) other*
- virtual bool [equals](#) ([Expr](#) \*e)=0
- virtual int [interp](#) ()=0
- virtual bool [has\\_variable](#) ()=0
- virtual [Expr](#) \* [subst](#) (std::string target, [Expr](#) \*other)=0

## 5.5.1 Constructor & Destructor Documentation

### 5.5.1.1 Var()

```
Var::Var (
    std::string value )
```

Sets String.

#### Parameters

<i>string</i>	value
---------------	-------

## 5.5.2 Member Function Documentation

### 5.5.2.1 equals()

```
bool Var::equals (
    Expr * other ) [virtual]
```

Compare Value.

#### Parameters

<i>Expr*</i>	other
--------------	-------

#### Returns

Boolean result

Implements [Expr](#).

### 5.5.2.2 has\_variable()

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

check if expression has variable

#### Returns

Boolean true

Implements [Expr](#).

### 5.5.2.3 interp()

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

Interprets the variable expression.

#### Returns

Runtime error as there is no value to interpret

Implements [Expr](#).

### 5.5.2.4 subst()

```
Expr * Var::subst (
    std::string target,
    Expr * other ) [virtual]
```

if variable matches target string, replaces variable with [Expr](#) other

#### Parameters

<i>string</i>	target - target string to replace
<i>Expr*</i>	other - replacement expression

#### Returns

if variable match, replace variable with [Expr](#) other; if not match, return variable

Implements [Expr](#).

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

- [/Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.h](#)
- [/Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.cpp](#)





## Chapter 6

# File Documentation

### 6.1 /Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/↵ ExpressionClass/cmdline.h File Reference

Contain functions to run program from terminal.

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

#### Functions

- void **use\_arguments** (int argc, char \*\*argv)

#### 6.1.1 Detailed Description

Contain functions to run program from terminal.

### 6.2 /Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/↵ ExpressionClass/cmdline.h

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

```
00001  
00006 #ifndef CMDLINE_H  
00007 #define CMDLINE_H  
00008  
00009 #define CATCH_CONFIG_RUNNER  
00010 #include "catch.h"  
00011 #include <iostream>  
00012 #include <cstdlib>  
00013  
00014 void use_arguments(int argc, char** argv);  
00015  
00016 #endif
```

### 6.3 /Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/↵ ExpressionClass/expr.cpp File Reference

Contains all the expression methods that are inherited by [Num](#), [Add](#), [Mult](#) and [Var](#).

```
#include "expr.h"
```

#### 6.3.1 Detailed Description

Contains all the expression methods that are inherited by [Num](#), [Add](#), [Mult](#) and [Var](#).

### 6.4 /Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/↵ ExpressionClass/expr.h File Reference

Contains the methods that would be defined in [expr.cpp](#).

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

#### Classes

- class [Expr](#)
- class [Num](#)
- class [Add](#)
- class [Mult](#)
- class [Var](#)

#### 6.4.1 Detailed Description

Contains the methods that would be defined in [expr.cpp](#).

### 6.5 /Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/↵ ExpressionClass/expr.h

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

```
00001
00006 #ifndef EXPR_H
00007 #define EXPR_H
00008
00009 #include <iostream>
00010 #include "catch.h"
00011
00012 class Expr {
00013     public:
00014         virtual bool equals(Expr *e) = 0;
00015         virtual int interp() = 0;
00016         virtual bool has_variable() = 0;
00017         virtual Expr* subst(std::string target, Expr* other) = 0;
```

```
00018 };
00019
00020 class Num : public Expr {
00021     private:
00022         int val;
00023     public:
00024         Num (int val);
00025         bool equals(Expr *other);
00026         int interp();
00027         bool has_variable();
00028         Expr* subst(std::string target, Expr* other);
00029 };
00030
00031 class Add : public Expr {
00032     private:
00033         Expr *lhs;
00034         Expr *rhs;
00035     public:
00036         Add(Expr *lhs, Expr *rhs);
00037         bool equals(Expr *other);
00038         int interp();
00039         bool has_variable();
00040         Expr* subst(std::string target, Expr* other);
00041 };
00042
00043 class Mult : public Expr {
00044     private:
00045         Expr *lhs;
00046         Expr *rhs;
00047     public:
00048         Mult(Expr *lhs, Expr *rhs);
00049         bool equals(Expr *other);
00050         int interp();
00051         bool has_variable();
00052         Expr* subst(std::string target, Expr* other);
00053 };
00054
00055 class Var : public Expr {
00056     private:
00057         std::string variable;
00058     public:
00059         Var(std::string value);
00060         bool equals(Expr *other);
00061         int interp();
00062         bool has_variable();
00063         Expr* subst(std::string target, Expr* other);
00064 };
00065
00066 #endif
```



# Index

[/Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/cmdline.h](#),  
[21](#)

[/Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.cpp](#),  
[22](#)

[/Users/jowietan/Desktop/MSD/CS6015/Homework/ExpressionClass/ExpressionClass/expr.h](#),  
[22](#)

Add, [9](#)

- Add, [9](#)
- equals, [10](#)
- has\_variable, [10](#)
- interp, [10](#)
- subst, [11](#)

equals

- Add, [10](#)
- Expr, [12](#)
- Mult, [14](#)
- Num, [16](#)
- Var, [18](#)

Expr, [11](#)

- equals, [12](#)
- has\_variable, [12](#)
- interp, [12](#)
- subst, [12](#)

has\_variable

- Add, [10](#)
- Expr, [12](#)
- Mult, [14](#)
- Num, [16](#)
- Var, [18](#)

interp

- Add, [10](#)
- Expr, [12](#)
- Mult, [14](#)
- Num, [16](#)
- Var, [18](#)

Mult, [13](#)

- equals, [14](#)
- has\_variable, [14](#)
- interp, [14](#)
- Mult, [13](#)
- subst, [14](#)

Num, [15](#)

- equals, [16](#)
- has\_variable, [16](#)
- interp, [16](#)

Num, [15](#)

- Expr, [12](#)
- subst, [16](#)

Num, [16](#)

- Add, [11](#)
- Expr, [12](#)
- Mult, [14](#)
- Num, [16](#)
- Var, [19](#)

Var, [17](#)

- equals, [18](#)
- has\_variable, [18](#)
- interp, [18](#)
- subst, [19](#)
- Var, [18](#)