

My Project

Generated by Doxygen 1.9.6

1 MSDScript -> CS6015_project	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 Add Class Reference	9
5.1.1 Constructor & Destructor Documentation	10
5.1.1.1 Add()	10
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 pretty_print_at()	11
5.1.2.5 print()	11
5.1.2.6 subst()	11
5.1.3 Member Data Documentation	12
5.1.3.1 lhs	12
5.1.3.2 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 pretty_print()	14
5.2.2.5 pretty_print_at()	14
5.2.2.6 print()	14
5.2.2.7 subst()	14
5.2.2.8 to_string()	14
5.2.2.9 to_string_pretty()	14
5.3 Mult Class Reference	15
5.3.1 Constructor & Destructor Documentation	15
5.3.1.1 Mult()	15
5.3.2 Member Function Documentation	16
5.3.2.1 equals()	16
5.3.2.2 has_variable()	16
5.3.2.3 interp()	16

5.3.2.4 pretty_print_at()	17
5.3.2.5 print()	17
5.3.2.6 subst()	17
5.3.3 Member Data Documentation	17
5.3.3.1 lhs	17
5.3.3.2 rhs	18
5.4 Num Class Reference	18
5.4.1 Constructor & Destructor Documentation	19
5.4.1.1 Num()	19
5.4.2 Member Function Documentation	19
5.4.2.1 equals()	19
5.4.2.2 has_variable()	19
5.4.2.3 interp()	20
5.4.2.4 pretty_print_at()	20
5.4.2.5 print()	20
5.4.2.6 subst()	20
5.4.3 Member Data Documentation	21
5.4.3.1 val	21
5.5 Var Class Reference	21
5.5.1 Constructor & Destructor Documentation	22
5.5.1.1 Var()	22
5.5.2 Member Function Documentation	22
5.5.2.1 equals()	22
5.5.2.2 has_variable()	23
5.5.2.3 interp()	23
5.5.2.4 pretty_print_at()	23
5.5.2.5 print()	23
5.5.2.6 subst()	23
5.5.3 Member Data Documentation	24
5.5.3.1 str	24
6 File Documentation	25
6.1 /Users/yutianqin/MSDScript/msd_cs6015_project/Expr.cpp File Reference	25
6.2 /Users/yutianqin/MSDScript/msd_cs6015_project/Expr.hpp File Reference	25
6.2.1 Enumeration Type Documentation	26
6.2.1.1 precedence_t	26
6.2.2 Function Documentation	26
6.2.2.1 use_arguments()	26
6.3 /Users/yutianqin/MSDScript/msd_cs6015_project/Expr.hpp	26
6.4 /Users/yutianqin/MSDScript/msd_cs6015_project/main.cpp File Reference	27
6.4.1 Macro Definition Documentation	28
6.4.1.1 CATCH_CONFIG_RUNNER	28

6.4.2 Function Documentation	28
6.4.2.1 main()	28
6.4.2.2 use_arguments()	28
6.5 /Users/yutianqin/MSDScript/msd_cs6015_project/subTest.cpp File Reference	28
6.5.1 Function Documentation	29
6.5.1.1 TEST_CASE()	29
6.6 /Users/yutianqin/MSDScript/msd_cs6015_project/Test.cpp File Reference	29
6.6.1 Function Documentation	29
6.6.1.1 TEST_CASE() [1/10]	29
6.6.1.2 TEST_CASE() [2/10]	30
6.6.1.3 TEST_CASE() [3/10]	30
6.6.1.4 TEST_CASE() [4/10]	30
6.6.1.5 TEST_CASE() [5/10]	30
6.6.1.6 TEST_CASE() [6/10]	30
6.6.1.7 TEST_CASE() [7/10]	30
6.6.1.8 TEST_CASE() [8/10]	30
6.6.1.9 TEST_CASE() [9/10]	31
6.6.1.10 TEST_CASE() [10/10]	31
Index	33

Chapter 1

MSDScript -> CS6015_project

Author

Yutian Qin

Date

02-07-2023

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Expr	12
Add	9
Mult	15
Num	18
Var	21

Chapter 3

Class Index

3.1 Class List

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

Add	9
Expr	12
Mult	15
Num	18
Var	21

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

/Users/yutianqin/MSDScript/msd_cs6015_project/ Expr.cpp	25
/Users/yutianqin/MSDScript/msd_cs6015_project/ Expr.hpp	25
/Users/yutianqin/MSDScript/msd_cs6015_project/ main.cpp	27
/Users/yutianqin/MSDScript/msd_cs6015_project/ subTest.cpp	28
/Users/yutianqin/MSDScript/msd_cs6015_project/ Test.cpp	29

Chapter 5

Class Documentation

5.1 Add Class Reference

```
#include <Expr.hpp>
```

Inheritance diagram for Add:



Public Member Functions

- [Add](#) ([Expr](#) *lhs, [Expr](#) *rhs)
- bool [equals](#) ([Expr](#) *expr) override
- int [interp](#) () override
- bool [has_variable](#) () override
- [Expr](#) * [subst](#) (std::string str, [Expr](#) *expr) override
- void [print](#) (std::ostream &os) override
- void [pretty_print_at](#) (std::ostream &os, [precedence_t](#) prec) override

Public Member Functions inherited from [Expr](#)

- virtual bool [equals](#) ([Expr](#) *expr)=0
- virtual int [interp](#) ()=0
- virtual bool [has_variable](#) ()=0
- virtual [Expr](#) * [subst](#) (std::string str, [Expr](#) *expr)=0
- virtual void [print](#) (std::ostream &os)=0
- std::string [to_string](#) ()
- void [pretty_print](#) (std::ostream &os, [precedence_t](#) prec)
- std::string [to_string_pretty](#) ()
- virtual void [pretty_print_at](#) (std::ostream &os, [precedence_t](#))=0

Public Attributes

- [Expr * lhs](#)
Left hand side.
- [Expr * rhs](#)
Right hand side.

5.1.1 Constructor & Destructor Documentation

5.1.1.1 Add()

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

5.1.2 Member Function Documentation

5.1.2.1 equals()

```
bool Add::equals (
    Expr * expr ) [override], [virtual]
```

Check whether the parameter equals to the [Add](#) object

Parameters

<i>expr</i>	
-------------	--

Returns

boolean

Implements [Expr](#).

5.1.2.2 has_variable()

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

The result should be true if the expression is a variable or contains a variable, false otherwise.

Returns

bool

Implements [Expr](#).

5.1.2.3 interp()

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

Interpret this [Add](#) object -> execute the calculation to add lhs and rhs of this [Add](#) object

Returns

sum of lhs and rhs

Implements [Expr](#).

5.1.2.4 pretty_print_at()

```
void Add::pretty_print_at (
    std::ostream & os,
    precedence_t prec ) [override], [virtual]
```

Implements [Expr](#).

5.1.2.5 print()

```
void Add::print (
    std::ostream & os ) [override], [virtual]
```

Implements [Expr](#).

5.1.2.6 subst()

```
Expr * Add::subst (
    std::string str,
    Expr * expr ) [override], [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>str</i>	
<i>expr</i>	

Returns

Expr*

Implements [Expr](#).

5.1.3 Member Data Documentation

5.1.3.1 lhs

[Expr*](#) Add::lhs

Left hand side.

5.1.3.2 rhs

[Expr*](#) Add::rhs

Right hand side.

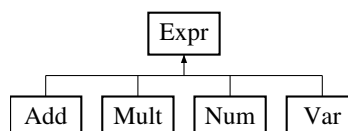
The documentation for this class was generated from the following files:

- /Users/yutianqin/MSDScript/msd_cs6015_project/[Expr.hpp](#)
- /Users/yutianqin/MSDScript/msd_cs6015_project/[Expr.cpp](#)

5.2 Expr Class Reference

```
#include <Expr.hpp>
```

Inheritance diagram for Expr:



Public Member Functions

- virtual bool [equals](#) ([Expr](#) *expr)=0
- virtual int [interp](#) ()=0
- virtual bool [has_variable](#) ()=0
- virtual [Expr](#) * [subst](#) (std::string str, [Expr](#) *expr)=0
- virtual void [print](#) (std::ostream &os)=0
- std::string [to_string](#) ()
- void [pretty_print](#) (std::ostream &os, [precedence_t](#) prec)
- std::string [to_string_pretty](#) ()
- virtual void [pretty_print_at](#) (std::ostream &os, [precedence_t](#))=0

5.2.1 Detailed Description

The parent class that will be inherited by all the other classes in this project

5.2.2 Member Function Documentation

5.2.2.1 [equals\(\)](#)

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

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

5.2.2.2 [has_variable\(\)](#)

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

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

5.2.2.3 [interp\(\)](#)

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

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

5.2.2.4 pretty_print()

```
void Expr::pretty_print (
    std::ostream & os,
    precedence_t prec )
```

5.2.2.5 pretty_print_at()

```
virtual void Expr::pretty_print_at (
    std::ostream & os,
    precedence_t ) [pure virtual]
```

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

5.2.2.6 print()

```
virtual void Expr::print (
    std::ostream & os ) [pure virtual]
```

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

5.2.2.7 subst()

```
virtual Expr * Expr::subst (
    std::string str,
    Expr * expr ) [pure virtual]
```

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

5.2.2.8 to_string()

```
std::string Expr::to_string ( )
```

5.2.2.9 to_string_pretty()

```
std::string Expr::to_string_pretty ( )
```

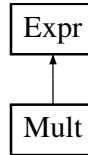
The documentation for this class was generated from the following files:

- [/Users/yutianqin/MSDScript/msd_cs6015_project/Expr.hpp](#)
- [/Users/yutianqin/MSDScript/msd_cs6015_project/Expr.cpp](#)

5.3 Mult Class Reference

```
#include <Expr.hpp>
```

Inheritance diagram for Mult:



Public Member Functions

- `Mult (Expr *lhs, Expr *rhs)`
- `bool equals (Expr *expr)` override
- `int interp ()` override
- `bool has_variable ()` override
- `Expr * subst (std::string, Expr *expr)` override
- `void print (std::ostream &os)` override
- `void pretty_print_at (std::ostream &os, precedence_t prec)` override

Public Member Functions inherited from Expr

- `virtual bool equals (Expr *expr)=0`
- `virtual int interp ()=0`
- `virtual bool has_variable ()=0`
- `virtual Expr * subst (std::string str, Expr *expr)=0`
- `virtual void print (std::ostream &os)=0`
- `std::string to_string ()`
- `void pretty_print (std::ostream &os, precedence_t prec)`
- `std::string to_string_pretty ()`
- `virtual void pretty_print_at (std::ostream &os, precedence_t)=0`

Public Attributes

- `Expr * lhs`
Left hand side.
- `Expr * rhs`
Right hand side.

5.3.1 Constructor & Destructor Documentation

5.3.1.1 Mult()

```

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

```

5.3.2 Member Function Documentation

5.3.2.1 equals()

```
bool Mult::equals (
    Expr * expr ) [override], [virtual]
```

Check whether the parameter equals to the [Mult](#) object

Parameters

<i>expr</i>	
-------------	--

Returns

boolean

Implements [Expr](#).

5.3.2.2 has_variable()

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

The result should be true if the expression is a variable or contains a variable, false otherwise.

Returns

bool

Implements [Expr](#).

5.3.2.3 interp()

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

Interpret this [Add](#) object -> execute the calculation that

Returns

product of lhs and rhs

Implements [Expr](#).

5.3.2.4 pretty_print_at()

```
void Mult::pretty_print_at (
    std::ostream & os,
    precedence_t prec ) [override], [virtual]
```

Implements [Expr](#).

5.3.2.5 print()

```
void Mult::print (
    std::ostream & os ) [override], [virtual]
```

Implements [Expr](#).

5.3.2.6 subst()

```
Expr * Mult::subst (
    std::string str,
    Expr * expr ) [override], [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>str</i>	
<i>expr</i>	

Returns

[Expr*](#)

Implements [Expr](#).

5.3.3 Member Data Documentation

5.3.3.1 lhs

```
Expr* Mult::lhs
```

Left hand side.

5.3.3.2 rhs

`Expr* Mult::rhs`

Right hand side.

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

- [/Users/yutianqin/MSDScript/msd_cs6015_project/Expr.hpp](#)
- [/Users/yutianqin/MSDScript/msd_cs6015_project/Expr.cpp](#)

5.4 Num Class Reference

```
#include <Expr.hpp>
```

Inheritance diagram for Num:



Public Member Functions

- [Num](#) (int *val*)
- bool [equals](#) ([Expr](#) *expr) override
- int [interp](#) () override
- bool [has_variable](#) () override
- [Expr](#) * [subst](#) (std::string str, [Expr](#) *expr) override
- void [print](#) (std::ostream &os) override
- void [pretty_print_at](#) (std::ostream &os, [precedence_t](#) prec) override

Public Member Functions inherited from [Expr](#)

- virtual bool [equals](#) ([Expr](#) *expr)=0
- virtual int [interp](#) ()=0
- virtual bool [has_variable](#) ()=0
- virtual [Expr](#) * [subst](#) (std::string str, [Expr](#) *expr)=0
- virtual void [print](#) (std::ostream &os)=0
- std::string [to_string](#) ()
- void [pretty_print](#) (std::ostream &os, [precedence_t](#) prec)
- std::string [to_string_pretty](#) ()
- virtual void [pretty_print_at](#) (std::ostream &os, [precedence_t](#))=0

Public Attributes

- int [val](#)

Value of this [Num](#) object.

5.4.1 Constructor & Destructor Documentation

5.4.1.1 Num()

```
Num::Num (
    int val ) [explicit]
```

5.4.2 Member Function Documentation

5.4.2.1 equals()

```
bool Num::equals (
    Expr * expr ) [override], [virtual]
```

Check whether the parameter equals to the [Num](#) object

Parameters

<i>expr</i>	
-------------	--

Returns

boolean

Implements [Expr](#).

5.4.2.2 has_variable()

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

Check whether this [Num](#) object is a variable

Returns

Implements [Expr](#).

5.4.2.3 interp()

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

Interpret this [Num](#) object -> get the actual value of this [Num](#)

Returns

value of this [Num](#) object

Implements [Expr](#).

5.4.2.4 pretty_print_at()

```
void Num::pretty_print_at (
    std::ostream & os,
    precedence_t prec ) [override], [virtual]
```

Implements [Expr](#).

5.4.2.5 print()

```
void Num::print (
    std::ostream & os ) [override], [virtual]
```

Implements [Expr](#).

5.4.2.6 subst()

```
Expr * Num::subst (
    std::string str,
    Expr * expr ) [override], [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>str</i>	
<i>expr</i>	

Returns

Expr*

Implements [Expr](#).

5.4.3 Member Data Documentation

5.4.3.1 val

```
int Num::val
```

Value of this [Num](#) object.

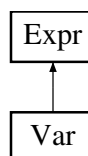
The documentation for this class was generated from the following files:

- [/Users/yutianqin/MSDScript/msd_cs6015_project/Expr.hpp](#)
- [/Users/yutianqin/MSDScript/msd_cs6015_project/Expr.cpp](#)

5.5 Var Class Reference

```
#include <Expr.hpp>
```

Inheritance diagram for Var:



Public Member Functions

- [Var](#) (std::string [str](#))
- bool [equals](#) ([Expr](#) *[expr](#)) override
- int [interp](#) () override
- bool [has_variable](#) () override
- [Expr](#) * [subst](#) (std::string, [Expr](#) *[expr](#)) override
- void [print](#) (std::ostream &[os](#)) override
- void [pretty_print_at](#) (std::ostream &[os](#), [precedence_t](#) [prec](#)) override

Public Member Functions inherited from [Expr](#)

- virtual bool [equals](#) ([Expr](#) *expr)=0
- virtual int [interp](#) ()=0
- virtual bool [has_variable](#) ()=0
- virtual [Expr](#) * [subst](#) (std::string str, [Expr](#) *expr)=0
- virtual void [print](#) (std::ostream &os)=0
- std::string [to_string](#) ()
- void [pretty_print](#) (std::ostream &os, [precedence_t](#) prec)
- std::string [to_string_pretty](#) ()
- virtual void [pretty_print_at](#) (std::ostream &os, [precedence_t](#))=0

Public Attributes

- std::string [str](#)
Value(String) of this [Var](#) object.

5.5.1 Constructor & Destructor Documentation

5.5.1.1 [Var\(\)](#)

```
Var::Var (
    std::string str ) [explicit]
```

constructors

5.5.2 Member Function Documentation

5.5.2.1 [equals\(\)](#)

```
bool Var::equals (
    Expr * expr ) [override], [virtual]
```

Check whether the parameter equals to the [Var](#) object

Parameters

expr	
----------------------	--

Returns

boolean

Implements [Expr](#).

5.5.2.2 has_variable()

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

The result should be true if the expression is a variable or contains a variable, false otherwise.

Returns

bool

Implements [Expr](#).

5.5.2.3 interp()

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

Show user an error message

Returns

error msg

Implements [Expr](#).

5.5.2.4 pretty_print_at()

```
void Var::pretty_print_at (
    std::ostream & os,
    precedence_t prec ) [override], [virtual]
```

Implements [Expr](#).

5.5.2.5 print()

```
void Var::print (
    std::ostream & os ) [override], [virtual]
```

Implements [Expr](#).

5.5.2.6 subst()

```
Expr * Var::subst (
    std::string string,
    Expr * expr ) [override], [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>str</i>	
<i>expr</i>	

Returns

Expr*

Implements [Expr](#).

5.5.3 Member Data Documentation

5.5.3.1 str

```
std::string Var::str
```

Value(String) of this [Var](#) object.

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

- [/Users/yutianqin/MSDScript/msd_cs6015_project/Expr.hpp](#)
- [/Users/yutianqin/MSDScript/msd_cs6015_project/Expr.cpp](#)

Chapter 6

File Documentation

6.1 /Users/yutianqin/MSDScript/msd_cs6015_project/Expr.cpp File Reference

```
#include "Expr.hpp"
#include <iostream>
#include <stdexcept>
#include <utility>
#include <sstream>
```

6.2 /Users/yutianqin/MSDScript/msd_cs6015_project/Expr.hpp File Reference

```
#include <iostream>
```

Classes

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

Enumerations

- enum [precedence_t](#) { [prec_none](#) = 0 , [prec_add](#) = 1 , [prec_mult](#) = 2 }

Functions

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

6.2.1 Enumeration Type Documentation

6.2.1.1 precedence_t

enum `precedence_t`

Enumerator

<code>prec_none</code>	
<code>prec_add</code>	
<code>prec_mult</code>	

6.2.2 Function Documentation

6.2.2.1 use_arguments()

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

6.3 /Users/yutianqin/MSDScript/msd_cs6015_project/Expr.hpp

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

```
00001
00002 #ifndef Expr_hpp
00003 #define Expr_hpp
00004 #include <iostream>
00005 #endif /* Expr_hpp */
00006
00007
00008
00009 void use_arguments( int argc, char **argv );
00010
00011
00012 typedef enum {
00013     prec_none = 0,      // = 0
00014     prec_add = 1,      // = 1
00015     prec_mult = 2      // = 2
00016 } precedence_t;
00017
00018
00019
00020
00021
00022
00023 class Expr {
00024 public:
00025     virtual bool equals( Expr *expr ) = 0;
00026     //HW03:
00027     virtual int interp() = 0;
00028     virtual bool has_variable() = 0;
00029     virtual Expr* subst( std::string str, Expr* expr ) = 0;
00030     //HW04:
00031     virtual void print( std::ostream& os ) = 0;
00032     std::string to_string();
00033     void pretty_print( std::ostream& os, precedence_t prec );
00034     std::string to_string_pretty();
00035     virtual void pretty_print_at( std::ostream& os, precedence_t ) = 0;
```



```

00036 };
00037
00038
00039
00040 class Num : public Expr {
00041 public:
00042     int val;
00043     explicit Num( int val );
00044     bool equals( Expr *expr ) override;
00045     //HW03:
00046     int interp() override;
00047     bool has_variable() override;
00048     Expr* subst( std::string, Expr* expr ) override;
00049     //HW04:
00050     void print( std::ostream& os ) override;
00051     void pretty_print_at( std::ostream& os, precedence_t prec ) override;
00052 };
00053
00054
00055
00056 class Add : public Expr {
00057 public:
00058     Expr *lhs;
00059     Expr *rhs;
00060     Add( Expr *lhs, Expr *rhs );
00061     bool equals( Expr *expr ) override;
00062     //HW03:
00063     int interp() override;
00064     bool has_variable() override;
00065     Expr* subst( std::string, Expr* expr ) override;
00066     //HW04:
00067     void print( std::ostream& os ) override;
00068     void pretty_print_at( std::ostream& os, precedence_t prec ) override;
00069 };
00070
00071
00072
00073 class Mult : public Expr {
00074 public:
00075     Expr *lhs;
00076     Expr *rhs;
00077     Mult( Expr *lhs, Expr *rhs );
00078     bool equals( Expr *expr ) override;
00079     //HW03:
00080     int interp() override;
00081     bool has_variable() override;
00082     Expr* subst( std::string, Expr* expr ) override;
00083     //HW04:
00084     void print( std::ostream& os ) override;
00085     void pretty_print_at( std::ostream& os, precedence_t prec ) override;
00086 };
00087
00088
00089
00090 class Var : public Expr {
00091 public:
00092     std::string str;
00093     explicit Var( std::string str );
00094     bool equals( Expr *expr ) override;
00095     //HW03:
00096     int interp() override;
00097     bool has_variable() override;
00098     Expr* subst( std::string, Expr* expr ) override;
00099     //HW04:
00100     void print( std::ostream& os ) override;
00101     void pretty_print_at( std::ostream& os, precedence_t prec ) override;
00102 };

```

6.4 /Users/yutianqin/MSDScript/msd_cs6015_project/main.cpp File Reference

```

#include "catch.h"
#include "Expr.hpp"

```

Macros

- #define CATCH_CONFIG_RUNNER

Functions

- int [main](#) (int argc, char **argv)
- void [use_arguments](#) (int argc, char **argv)

6.4.1 Macro Definition Documentation

6.4.1.1 CATCH_CONFIG_RUNNER

```
#define CATCH_CONFIG_RUNNER
```

6.4.2 Function Documentation

6.4.2.1 main()

```
int main (  
    int argc,  
    char ** argv )
```

6.4.2.2 use_arguments()

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

6.5 /Users/yutianqin/MSDScript/msd_cs6015_project/subTest.cpp File Reference

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

Functions

- [TEST_CASE](#) ("print")

6.5.1 Function Documentation

6.5.1.1 TEST_CASE()

```
TEST_CASE (
    "print" )
```

Thomas' test

6.6 /Users/yutianqin/MSDScript/msd_cs6015_project/Test.cpp File Reference

```
#include "Expr.hpp"
#include "catch.h"
#include <iostream>
#include <sstream>
```

Functions

- [TEST_CASE](#) ("pretty_print & to_string_pretty")
- [TEST_CASE](#) ("print & to_string")
- [TEST_CASE](#) ("equals")
- [TEST_CASE](#) ("interp")
- [TEST_CASE](#) ("subst")
- [TEST_CASE](#) ("Kevin Tests [Num](#) Equality")
- [TEST_CASE](#) ("Kevin Tests [Add](#) Equality")
- [TEST_CASE](#) ("Kevin Tests MultExpr Equality")
- [TEST_CASE](#) ("Kevin Tests [Var](#) Equality")
- [TEST_CASE](#) ("Kevin Tests Interpret Tests")

6.6.1 Function Documentation

6.6.1.1 TEST_CASE() [1/10]

```
TEST_CASE (
    "equals" )
```

6.6.1.2 TEST_CASE() [2/10]

```
TEST_CASE (
    "interp" )
```

6.6.1.3 TEST_CASE() [3/10]

```
TEST_CASE (
    "Kevin Tests Add Equality" )
```

6.6.1.4 TEST_CASE() [4/10]

```
TEST_CASE (
    "Kevin Tests Interpret Tests" )
```

6.6.1.5 TEST_CASE() [5/10]

```
TEST_CASE (
    "Kevin Tests MultExpr Equality" )
```

6.6.1.6 TEST_CASE() [6/10]

```
TEST_CASE (
    "Kevin Tests Num Equality" )
```

6.6.1.7 TEST_CASE() [7/10]

```
TEST_CASE (
    "Kevin Tests Var Equality" )
```

6.6.1.8 TEST_CASE() [8/10]

```
TEST_CASE (
    "pretty_print & to_string_pretty" )
```

6.6.1.9 TEST_CASE() [9/10]

```
TEST_CASE (
    "print & to_string" )
```

6.6.1.10 TEST_CASE() [10/10]

```
TEST_CASE (
    "subst" )
```


Index

/Users/yutianqin/MSDScript/msd_cs6015_project/Expr.cpp, [Expr](#), [13](#)
[25](#)
/Users/yutianqin/MSDScript/msd_cs6015_project/Expr.hpp, [Num](#), [19](#)
[25](#) [Var](#), [23](#)
/Users/yutianqin/MSDScript/msd_cs6015_project/Test.cpp, [interp](#)
[29](#)
/Users/yutianqin/MSDScript/msd_cs6015_project/main.cpp, [Add](#), [11](#)
[27](#) [Expr](#), [13](#)
/Users/yutianqin/MSDScript/msd_cs6015_project/subTest.cpp, [Mult](#), [16](#)
[28](#) [Num](#), [19](#)
[Var](#), [23](#)

[Add](#), [9](#)
 [Add](#), [10](#)
 [equals](#), [10](#)
 [has_variable](#), [10](#)
 [interp](#), [11](#)
 [lhs](#), [12](#)
 [pretty_print_at](#), [11](#)
 [print](#), [11](#)
 [rhs](#), [12](#)
 [subst](#), [11](#)

[CATCH_CONFIG_RUNNER](#)
 [main.cpp](#), [28](#)

[equals](#)
 [Add](#), [10](#)
 [Expr](#), [13](#)
 [Mult](#), [16](#)
 [Num](#), [19](#)
 [Var](#), [22](#)

[Expr](#), [12](#)
 [equals](#), [13](#)
 [has_variable](#), [13](#)
 [interp](#), [13](#)
 [pretty_print](#), [13](#)
 [pretty_print_at](#), [14](#)
 [print](#), [14](#)
 [subst](#), [14](#)
 [to_string](#), [14](#)
 [to_string_pretty](#), [14](#)

[Expr.hpp](#)
 [prec_add](#), [26](#)
 [prec_mult](#), [26](#)
 [prec_none](#), [26](#)
 [precedence_t](#), [26](#)
 [use_arguments](#), [26](#)

[has_variable](#)
 [Add](#), [10](#)

[lhs](#)
 [Add](#), [12](#)
 [Mult](#), [17](#)

[main](#)
 [main.cpp](#), [28](#)

[main.cpp](#)
 [CATCH_CONFIG_RUNNER](#), [28](#)
 [main](#), [28](#)
 [use_arguments](#), [28](#)

[Mult](#), [15](#)
 [equals](#), [16](#)
 [has_variable](#), [16](#)
 [interp](#), [16](#)
 [lhs](#), [17](#)
 [Mult](#), [15](#)
 [pretty_print_at](#), [16](#)
 [print](#), [17](#)
 [rhs](#), [17](#)
 [subst](#), [17](#)

[Num](#), [18](#)
 [equals](#), [19](#)
 [has_variable](#), [19](#)
 [interp](#), [19](#)
 [Num](#), [19](#)
 [pretty_print_at](#), [20](#)
 [print](#), [20](#)
 [subst](#), [20](#)
 [val](#), [21](#)

[prec_add](#)
 [Expr.hpp](#), [26](#)

[prec_mult](#)
 [Expr.hpp](#), [26](#)

[prec_none](#)
 [Expr.hpp](#), [26](#)

[precedence_t](#)
 [Expr.hpp](#), [26](#)

- pretty_print
 - Expr, [13](#)
- pretty_print_at
 - Add, [11](#)
 - Expr, [14](#)
 - Mult, [16](#)
 - Num, [20](#)
 - Var, [23](#)
- print
 - Add, [11](#)
 - Expr, [14](#)
 - Mult, [17](#)
 - Num, [20](#)
 - Var, [23](#)
- rhs
 - Add, [12](#)
 - Mult, [17](#)
- str
 - Var, [24](#)
- subst
 - Add, [11](#)
 - Expr, [14](#)
 - Mult, [17](#)
 - Num, [20](#)
 - Var, [23](#)
- subTest.cpp
 - TEST_CASE, [29](#)
- Test.cpp
 - TEST_CASE, [29–31](#)
- TEST_CASE
 - subTest.cpp, [29](#)
 - Test.cpp, [29–31](#)
- to_string
 - Expr, [14](#)
- to_string_pretty
 - Expr, [14](#)
- use_arguments
 - Expr.hpp, [26](#)
 - main.cpp, [28](#)
- val
 - Num, [21](#)
- Var, [21](#)
 - equals, [22](#)
 - has_variable, [23](#)
 - interp, [23](#)
 - pretty_print_at, [23](#)
 - print, [23](#)
 - str, [24](#)
 - subst, [23](#)
 - Var, [22](#)