ReVision

Olaf Bernstein

June 1, 2018

Contents

1 Wishlist

1.1 Keywords

keyword	description

use	You will be able to list specific variables you want to use and the scope of the next code block gets reduced to this variables. Should be able to return values, the return type will get deduced at compile time.
	`a = use x, y {`
	`        ...`
	`        // return from the use block`
	`        return 3;`
	`};`

namespace	C++ Style namespace keyword.

1.2 Type System

The type system will be strongly typed with something similar to the C++ keyword *auto* and will use static type checking.

1.2.1 Primitive Types

signed	usigned
—	char
int8	uint8
int16	uint16
int32	uint32
int64	uint64
float	—
double	—

1.2.2 Composite Types

type	description
union	Like the C/C++ union type.
struct	Like the C struct.
class	C++ class type with some changes:
	Private members variables are accessible but can't be changed.
	Functions must be separated from the member variables.

1.2.3 Type Qualifier

These work for all types above:

qualifier	description
const	Like the C/C++ const type qualifier.

1.2.4 Other

keyword/operator	description
'*' type	Like the C/C++ pointer.
'&' type	Like the C/C++ reference.

1.3 Functions

Can be declared in the global namespace but also inside a function which will only be accessible in the local space. Forward declarations similar to C/C++ will be supported and operator overloading.

1.4 Operator