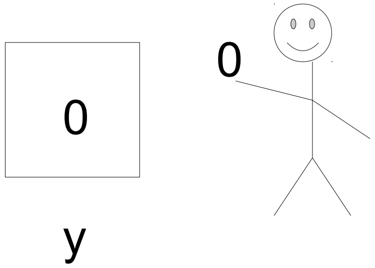
Lvalues and Rvalues

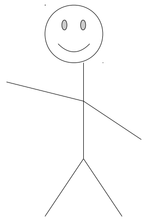
```
int &y = 0;
int x = y;
const SymOp &s1= SymOp();
```

Lvalue denotes an object whose resource cannot be reused and are generally the names of objects.



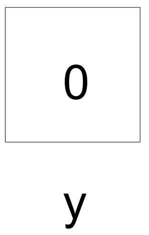
Rvalues are objects that cannot be reused. Usually they are temporary/soon to be destroyed.

Lvalues and Rvalues



```
int &y = 0;
int x = y;
const SymOp &s1= SymOp();
```

Lvalue denotes an object whose resource cannot be reused and are generally the names of objects.



0

y

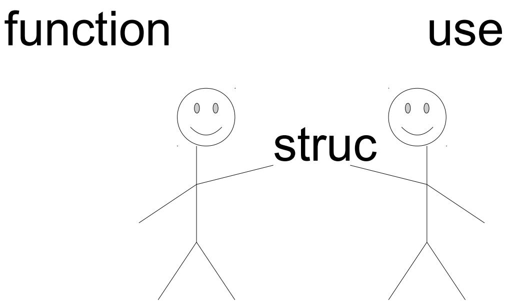
Rvalues are objects that cannot be reused. Usually they are temporary/soon to be destroyed.

Lvalue Reference vs. Rvalue Reference

```
 void apply_sym(const SymOp& op, Structure& struc);
Structure apply_sym(const SymOp& op, const Structure &struc);
Structure apply_sym(const SymOp& op, Structure &&struc);
```

```
void apply_sym(const SymOp& op, Structure& struc);
```

Applies symmetry operation, op, in a non-const way
using struc as a bound lvalue reference.
Hint look for single &.



Lvalue Reference vs. Rvalue Reference

```
 void apply_sym(const SymOp& op, Structure& struc);
Structure apply_sym(const SymOp& op, const Structure &struc);
Structure apply_sym(const SymOp& op, Structure &&struc);
```

```
void apply_sym(const SymOp& op, Structure& struc);
```

Applies symmetry operation, op, in a non-const way
using struc as a bound lvalue reference.
Hint look for single &.

function

user

struc

struc

Lvalue Reference vs. Rvalue Reference

```
 void apply_sym(const SymOp& op, Structure& struc);
Structure apply_sym(const SymOp& op, const Structure &struc);
Structure apply_sym(const SymOp& op, Structure &&struc);
```

```
void apply_sym(const SymOp& op, Structure& struc);
```

Applies symmetry operation, op, in a non-const way
using struc as a bound lvalue reference.
Hint look for single &.

Lvalue Reference vs. Rvalue Reference

```
 void apply_sym(const SymOp& op, Structure& struc);
Structure apply_sym(const SymOp& op, const Structure &struc);
Structure apply_sym(const SymOp& op, Structure &&struc);
```

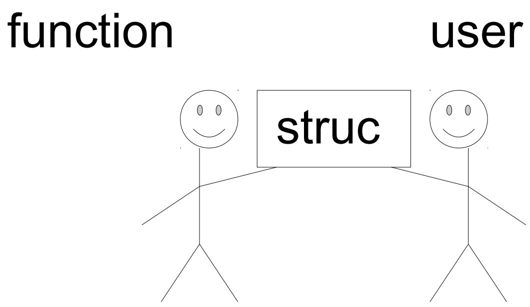
```
void apply_sym(const SymOp& op, Structure& struc);
```

Applies symmetry operation, op, in a non-const way
using struc as a bound lvalue reference.
Hint look for single &.

Lvalue Reference vs. Rvalue Reference

```
 void apply_sym(const SymOp& op, Structure& struc);
Structure apply_sym(const SymOp& op, const Structure &struc);
Structure apply_sym(const SymOp& op, Structure &&struc);
```

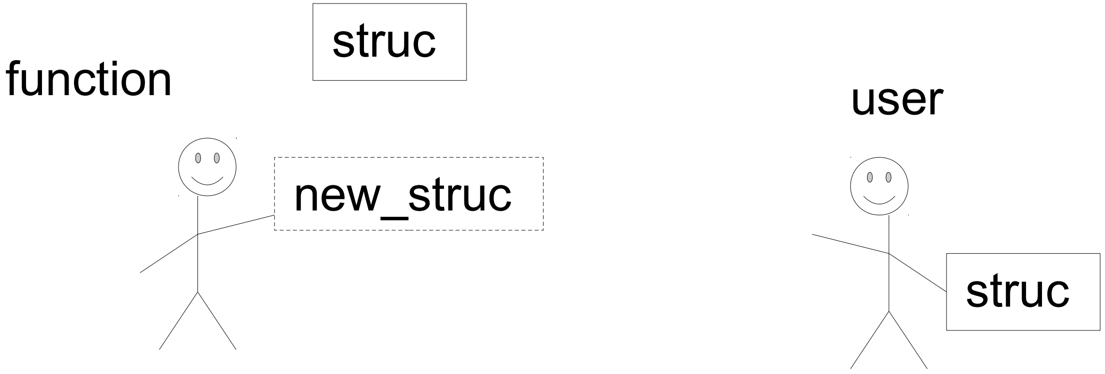
Structure apply_sym(const SymOp& op, const Structure &struc);
Applies symmetry operation, op, in a secure way using struc as a bound const lvalue reference.
Hint look for single & and const.



Lvalue Reference vs. Rvalue Reference

```
 void apply_sym(const SymOp& op, Structure& struc);
Structure apply_sym(const SymOp& op, const Structure &struc);
Structure apply_sym(const SymOp& op, Structure &&struc);
```

Structure apply_sym(const SymOp& op, const Structure &struc);
Applies symmetry operation, op, in a secure way using struc as a bound const lvalue reference.
Hint look for single & and const.



function

struc

new_struc

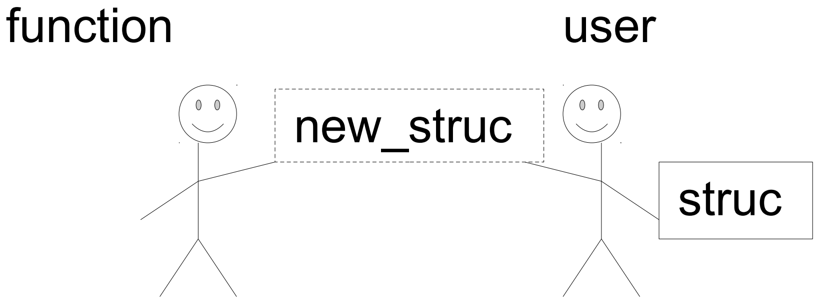
user

struc

Lvalue Reference vs. Rvalue Reference

```
 void apply_sym(const SymOp& op, Structure& struc);
Structure apply_sym(const SymOp& op, const Structure &struc);
Structure apply_sym(const SymOp& op, Structure &&struc);
```

Structure apply_sym(const SymOp& op, const Structure &struc);
Applies symmetry operation, op, in a secure way using struc as a bound const lvalue reference.
Hint look for single & and const.



function

user

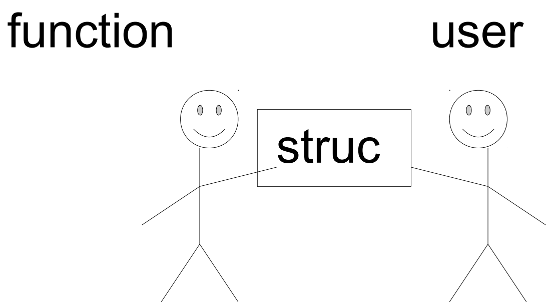
new_struc

struc

Lvalue Reference vs. Rvalue Reference

```
 void apply_sym(const SymOp& op, Structure& struc);
Structure apply_sym(const SymOp& op, const Structure &struc);
Structure apply_sym(const SymOp& op, Structure &&struc);
```

Structure apply_sym(const SymOp& op, Structure &&struc);
Applies symmetry operation, op, in a potentially non-const way using struc as a bound rvalue reference.
Hint look for double &&.

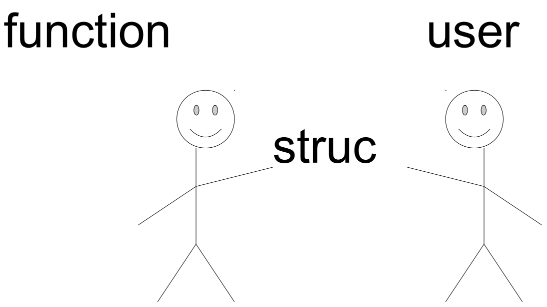


Lvalue Reference vs. Rvalue Reference

```
 void apply_sym(const SymOp& op, Structure& struc);
Structure apply_sym(const SymOp& op, const Structure &struc);
Structure apply_sym(const SymOp& op, Structure &&struc);
```

```
Structure apply_sym(const SymOp& op, Structure &&struc);
```

Applies symmetry operation, op, in a potentially non-const way using struc as a bound rvalue reference.
Hint look for double &&.



function

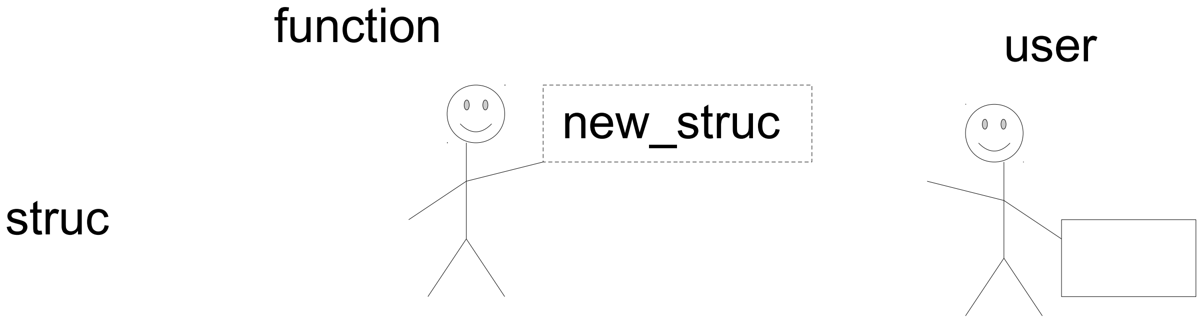
user

struc

Lvalue Reference vs. Rvalue Reference

```
 void apply_sym(const SymOp& op, Structure& struc);
Structure apply_sym(const SymOp& op, const Structure &struc);
Structure apply_sym(const SymOp& op, Structure &&struc);
```

Structure apply_sym(const SymOp& op, Structure &&struc);
Applies symmetry operation, op, in a potentially non-const
way using struc as a bound rvalue reference.
Hint look for double &&.



Lvalue Reference vs. Rvalue Reference

```
 void apply_sym(const SymOp& op, Structure& struc);
Structure apply_sym(const SymOp& op, const Structure &struc);
Structure apply_sym(const SymOp& op, Structure &&struc);
```

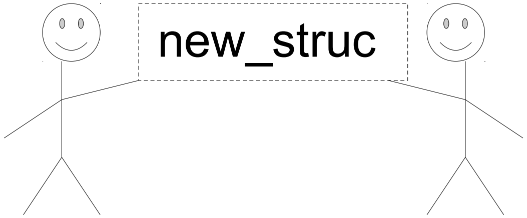
```
Structure apply_sym(const SymOp& op, Structure &&struc);
```

Applies symmetry operation, op, in a potentially non-const way using struc as a bound rvalue reference.

Hint look for double &&.

function

user



new_struc

struc

Rules

- Cannot bind rvalue reference to lvalue or lvalue reference
- Cannot bind non-const lvalue reference to rvalue or rvalue reference