How to check an identifier (a variable or a number etc) is left or right value in C++?

[intro]

forward

The func forward returns a rvalue(or right value) iff the argument is rvalue.

[namespace]

std

[library]

<utility>

[P.S.]

(1)lvalue and rvalue

lvalue: left value, the identifier which is left to assignment symbol.

rvalue: right value, the identifier which is right to assignment symbol.

Consider the following statements.

x is lvalue and 5 is rvalue.

y is lvalue and 3 is rvalue.

x is lvalue and y+1 is rvalue.

[code]

#include <utility>

#include <iostream>

using namespace std;

// function with lvalue and rvalue reference overloads:

void overloaded (const int& x)

{

std::cout << "[lvalue]";

}

void overloaded (int&& x)

{

std::cout << "[rvalue]";

}

// function template taking rvalue reference to deduced type:

template <class T> void fn (T&& x)

{

overloaded (x); // always an lvalue

overloaded (std::forward<T>(x)); // rvalue if argument is rvalue

}

int main ()

{

int a;

cout << "calling fn with lvalue: ";

fn (a);

cout << '\n';

cout << "calling fn with rvalue: ";

fn (0);

cout << '\n';

return 0;

}

[result]

calling fn with lvalue: [lvalue][lvalue]

calling fn with rvalue: [lvalue][rvalue]

[ref]