Unary promote +operator

Đorđe Anđelković

Definition

The built-in unary plus operator returns the value of its operand. The only situation where it is not a no-op is when the operand has integral type or unscoped enumeration type, which is changed by integral promotion, e.g, it converts char to int or if the operand is subject to lvalue-to-rvalue, array-to-pointer, or function-to-pointer conversion.

First example

```
#include <iostream>

template<typename T>
void foo(T &&t)
{
    t++;
    std::cout << t << std::endl;
}

int main()
{
    int x = 55;
    foo(+x);
    std::cout << x << std::endl;
}
```

Second example

```cpp
#include <iostream>
#include <vector>

template<typename T>
void printRange(T const &begin, T const &end)
{
    for (T it = begin; it != end; it++) {
        std::cout << *it << " ";
    }
    std::cout << std::endl;
}

int main()
{
    std::vector<int> V{1, 3, 5, 7, 9};
    int A[10] = {2, 4, 6, 8, 10, 12};

    printRange(V.begin(), V.end());
    // printRange(A,A+5); // compile error
}
```

Second example

```cpp
#include <iostream>
#include <vector>

template<typename T>
void printRange(T const &begin, T const &end)
{
    for (T it = begin; it != end; it++) {
        std::cout << *it << " ";
    }
    std::cout << std::endl;
}

int main()
{
    std::vector<int> V{1, 3, 5, 7, 9};
    int A[10] = {2, 4, 6, 8, 10, 12};

    printRange(V.begin(), V.end());
    printRange(+A, +A + 5);
}
```

Third example

```cpp
#include <iostream>

struct Foo
{
    static int const value = 42;
};

template<typename T>
void f(T const &t)
{
    std::cout << t << std::endl;
}

int main()
{
    // fails to link - tries to get the address of "Foo::value"!
    // f(Foo::value);

    // works - pass a temporary value
    f(+Foo::value);
}
```

The end

Reference:

https://en.cppreference.com/w/cpp/language/operator_arithmetic#Unary_arithmetic_operators

Godbolt:

first example: https://godbolt.org/z/K45vKMWqd

unary plus: https://godbolt.org/z/TPW75e3fq

static int: https://godbolt.org/z/oMdcEjzo9