Add a New Keyword `undecl`

Document #:	P1234R0
Date:	2025-12-08
Project:	Programming Language C++
Audience:	EWG CWG

Reply-to:	Wang Jiafeng <wjf@zenkee.cn> Wang Hanzhi <wanghzh2021@163.com>

1 Abstract

This paper proposes the new keyword `undecl` to explicitly end the lifetime of a previously declared variable, allowing the same name to be reused and enabling the compiler to reclaim the associated memory storage or register.

2 Introduction

In classic C, variables could be declared only at the beginning of a function. C++ removed this restriction and allows declarations anywhere inside a function, with lifetime starting at the point of declaration. However, the end of a variable's lifetime is still determined by the closing brace of its enclosing scope. For short-lived temporaries, the common workaround is to introduce an extra pair of braces, but this is indirect and forces any longer-lived variables used later to be hoisted outside those braces. An `undecl` statement that ends a variable's lifetime in place would be a cleaner solution.

3 Proposed Solution

Traditional

```
int main() {
    int b;// must be declared
          // outside the extra braces
    {
        int a = 3;
        // ...
        b = 4;
    }// 'a' silently goes out of scope
    char* a;
}
```

Proposed

```
int main() {
    int a = 3;
    // ...
    int b = calc();
    // ...
    undecl a;// end lifetime of 'a', must
             // appear at the same brace
             // level as the declaration.

    char* a;// reuse the name
            // for a different type
}
```

If `b` is a complex object, hoisting it may incur performance penalties.

4 Impact on the Standard

No breaking changes to the rest of the language.

5 Proposed Wording

TBD

6 Implementation Experience

TBD