C++ Allows Variable Declarations Between Statements

To address the problem, **C++**
- **allows variable declarations**
- to be **interleaved with statements**
- (later, this feature was back-propagated into **C**).

`for` loop iteration variables
- of a single base type
- can also be declared in the init clause
- (not supported for other loops).

Avoiding Extra Work with Conditionals Still Tricky

The problem is not completely solved:
consider the code below.

```
int choice = some_calculation ();
if (choice) {
    // use constructor 1 for var
} else {
    // use constructor 2 for var
}
// use var ...
```


Op-Assign Operators Should Be Used When Possible

Use of op-assign operators
- +=, -=, *=, and so forth
- is also **encouraged in C++**.

Why?

Are these all the same?

```
A = A + B
A = B + A
  A += B
```

One answer: not if + is string concatenation.

Compilers May Fail to Perform "Obvious" Optimizations

Even restricting our attention to `A = A + B`,
proving equivalence may be challenging.

A **compiler**
- **must consider aliasing**:
- **can A be changed before** the **sum operation is complete**?

Compiler can analyze the problem
- if both functions, and
- may be able to transform into `A += B`.

In contrast, **programmer must handle aliasing** when implementing **`operator+=`**.

