C++

```cpp
int c;

int main()
{
    int c = 100;
    if (c > 0)
    {
        int c = 200;
        cout << c;
    }
    cout << c;
    cout << c;
}
```

Java

```java
public static int function()
{
    int i, j;
    no confusion
    no globals
    no ::
    for (int j = 0; ... )
    {...
        j dies here
    }
}
Are we referring to the same place?

Address Copies

\[ \text{If } (\text{code}1 = 0 \text{df2}) \]

Pointer Arith.

What is this?

\[ 0 \text{df}1 = 0 \text{df2} \]

Referring to \text{df}1 + \text{df}2;

\[ 1 \text{df}1 = \text{new } 1 \text{st c} \]

\[ 1 \text{df}1 + \text{df}2 \]

\[ \text{referring} \]

Referring to \text{df}1 + \text{df}1.
C#  
\[ \text{if (response == 'N')} \]
\[ \text{Not zero} \]
\[ \text{N \rightarrow True \}} \text{ True Result} \]
\[ \text{if ('N' \in \text{response})} \]

Java

\[ \text{ints are not bools} \]
\[ \text{error} \]

\[ \text{if (current != null)} \]
\[ \text{spell if all out} \]
C++

int * array;

Java

int[] array = new int[size];

NOT Supported
Sorry. Not supported.
```csharp
public void copy (node node) {
    if (node == null) {
        return;
    } else {
        newNode = new node (node.source);
        if (node == source) {
            newNode = new node (node.source);
        } else {
            newNode.left = copy (node.left);
            newNode.right = copy (node.right);
        }
        return newNode;
    }
}
```