Initialization & Assignment

We know there is a difference between `==` and `=`

But there are two kinds of `=`:

- `=` in a declaration, like `int a = 4;` (initialization)
- `=` elsewhere, like `a = 4;` (assignment)

Initialization & Assignment

```cpp
// complex.h

#ifndef COMPLEX_H__
#define COMPLEX_H__
#include <iostream>

class Complex {
public:
    // Default constructor
    Complex() : Complex(0.0, 0.0) { std::cout << "Default" << std::endl; }
    // Non-default constructor
    Complex(double r, double i) : real(r), imag(i) { std::cout << "Non-default" << std::endl; }
    // Copy constructor
    Complex(const Complex& c) : real(c.real), imag(c.imag) { std::cout << "Copy" << std::endl; }
    // Assignment operator
    Complex& operator=(const Complex& rhs) {
        std::cout << "Assign" << std::endl;
        real = rhs.real;
        imag = rhs.imag;
        return *this;
    }

    double get_real() const { return real; }
    double get_imag() const { return imag; }

private:
    double real, imag;
};
#endif // COMPLEX_H__
```

Initialization & Assignment

```cpp
// complex_main.cpp
#include "complex.h"

int main() {
    Complex c;                 //default (no-argument) constructor call
    Complex c2 = {4.9, 0.5};   // = in declaration: *initialization*
    Complex c3 = c;            // = in declaration; also *initialization*
    c3 = c2;                   // = outside declaration: *assignment*
    if(c3.get_real() == 4.9) { // == is *equality testing*
        std::cout << "Real part of c3 is equal to 4.9" << std::endl;
    }
    return 0;
}
```

```
$ g++ -o complex_main complex_main.cpp -std=c++11 -pedantic -Wall -Wextra
$ ./complex_main
Non-default
Default
Non-default
Copy
Assign
Real part of c3 is equal to 4.9
```

Quiz!

What output is printed by the following program?

```cpp
#include <iostream>

int x = 0;

class Foo {
public:
    Foo() { x++; }
    Foo(const Foo &obj) { x += 3; }
    Foo& operator=(const Foo &obj){
        x += 2; return *this;
    }
};

int main() {
    Foo f1;
    Foo f2(f1);
    Foo f3 = f1;
    f3 = f1;
    std::cout << x << std::endl;
    return 0;
}
```

A. 0

B. 4

C. 8

D. 9

E. Some other value is printed

Quiz - answers

What output is printed by the following program?

```
 1  #include <iostream>
 2
 3  int x = 0;
 4
 5  class Foo {
 6  public:
 7      Foo() { x++; }
 8      Foo(const Foo &obj) { x += 3; }
 9      Foo& operator=(const Foo &obj){
10          x += 2; return *this;
11      }
12  };
13
14  int main() {
15      Foo f1;
16      Foo f2(f1);
17      Foo f3 = f1;
18      f3 = f1;
19      std::cout << x << std::endl;
20      return 0;
21  }
```

At line 19:

Symbols (Scope)	Values
x(global)	9