public static boolean isEven(int x) {
    return x % 2 == 0;
}

public static int triple(int x) {
    return x * 3;
}

public static int someFunc(int x) {
    if (isEven(x)) {
        return x + 1;
    } else {
        return triple(x);
    }
}

public static void main(String args[]) {
    int x = 5;
    System.out.println(someFunc(x));
}
Why is this ok? different scope. different vars, same name

What Happens Here?

```java
public static int f(int x) {
    if (x==1) {
        return 1;
    } else {
        return 2*f(x-1);
    }
}
public static void main(String args[]) {
    int x = 3;
    System.out.println(f(x));
}
```

How Do We Handle This?

It’s no different from any other function call you’ve ever made:

- copy the arguments to the function
- execute the function
- jump back

Is this even legal?

- **Question:** Is it ok for a function to call itself?
- **Answer:** Yes. We call it **recursion**
It's almost as though we did this
(but don’t actually recopy the function. This would be an error)
public class SimpleRecur2 {
    public static int f(int x) {
        if (x==1) {
            return 2;
        } else {
            return 2*f(x-1)+1;
        }
    }
    public static void main(String args[]) {
        int x = 5;
        System.out.println(f(x));
    }
}