Lecture 7 - Sep 25

OOP Review

Caller vs. Callee
Tracing Chain of Method Calls via a Stack
Catch-or-Specify Requirement
To Handle or Not to Handle: V1

Announcements/Reminders

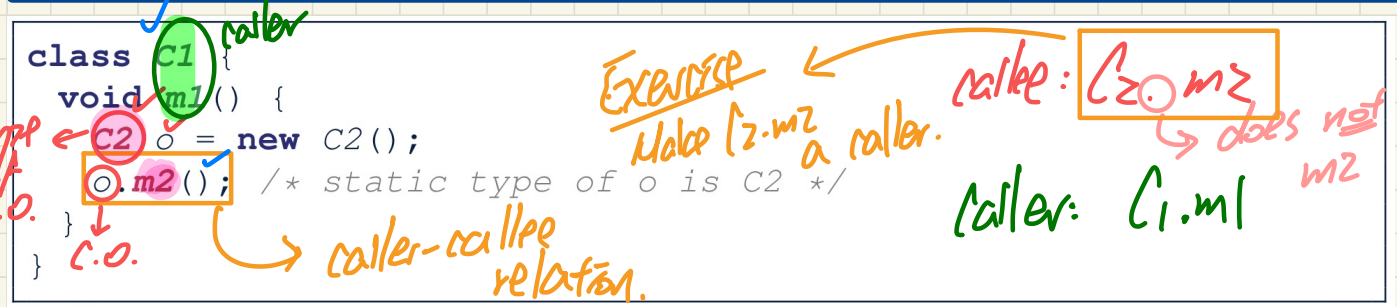
- Today's class: notes template posted
- Priorities:
  + Review Lab0
  + Complete Lab1; Due: Next Tuesday (Sep 30)

method that calls another method → callee

Caller vs. **Callee** method that's called by another method → caller.

* Each method call defines a caller-callee relation.

- caller is the client using the service provided by another method.
- callee is the supplier providing the service to another method.

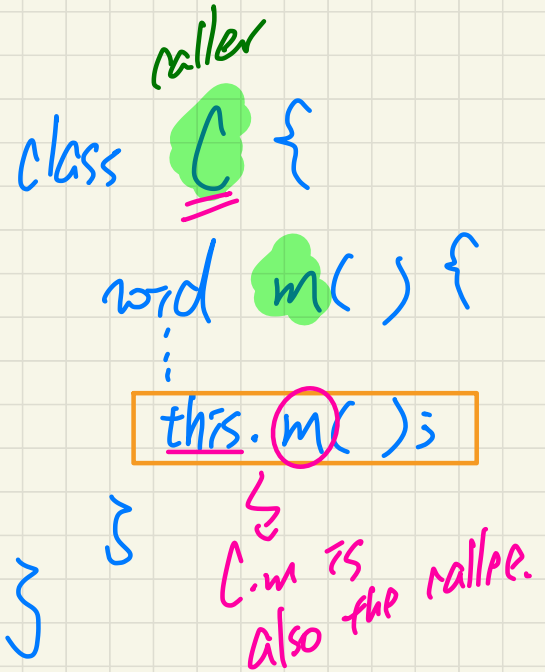


Q: Can a method be a caller and a callee simultaneously?

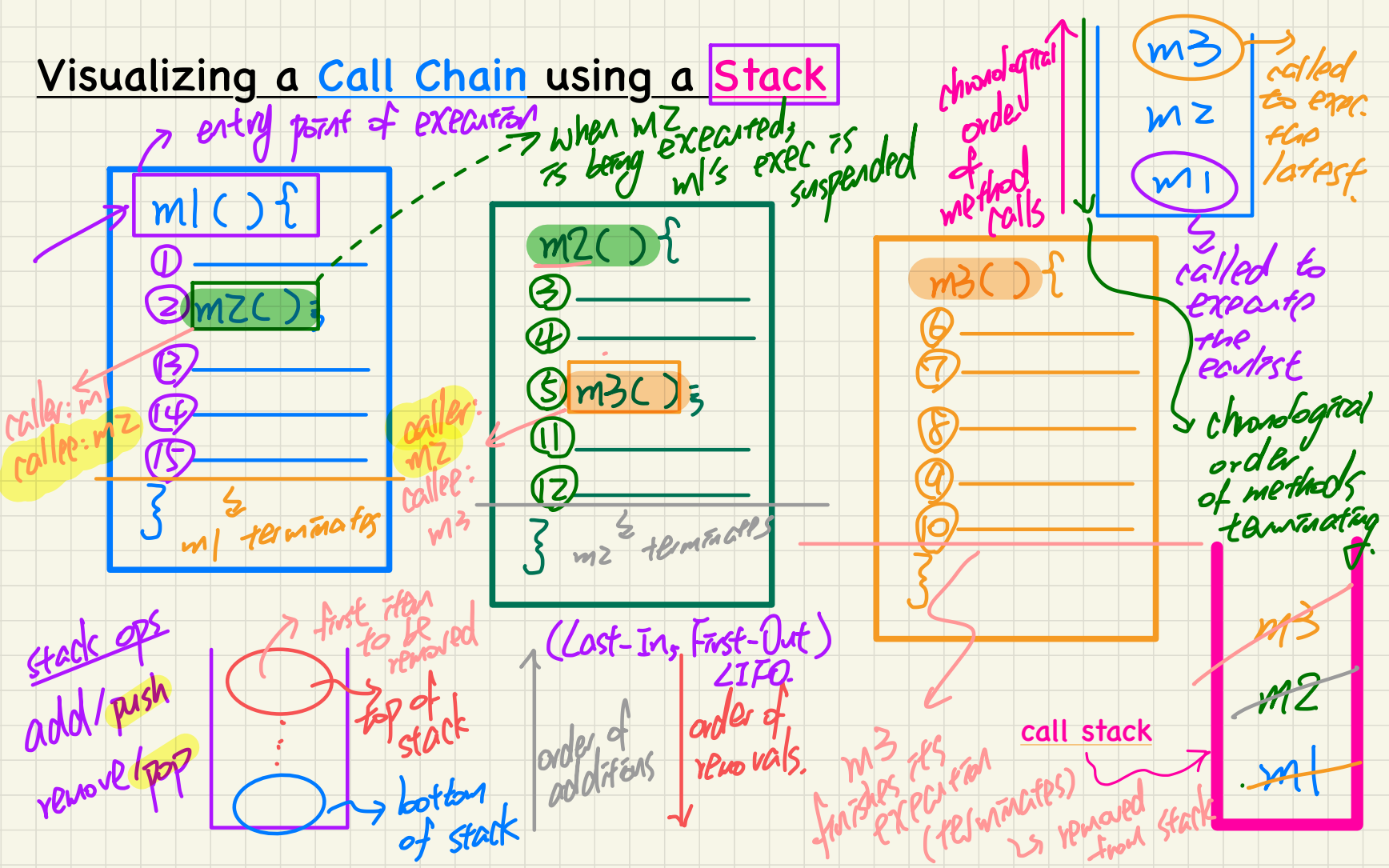
(Alt 1) Make C1.m1 a callee.

```
class C3 {
  void m() { C1 o = new C1(); o.m1(); }
}
```

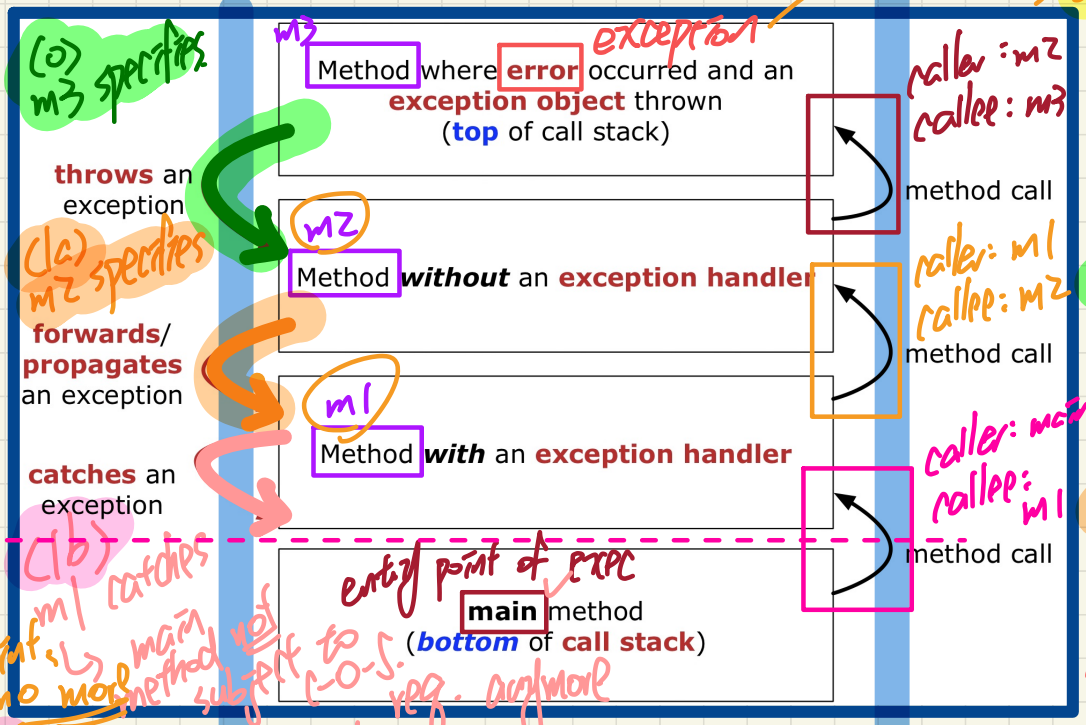
callee


caller
class C {
void m( ) {
this.m( );
}
}
C.m is
also the callee.

Visualizing a Call Chain using a Stack



What to Do When an Exception is Thrown: Call Stack



(b) m1 catches

beyond this point, no more main method subject to C-O-S. reg. anymore

as soon as an exception object is thrown, the call stack steps going.

Catch-or-Specify Req.

(a) Where the exception is originated, specify it.

(1a) A method may specify the exceptions which is propagated to its caller.

(1b) A method may catch the exception.

↳ beyond this method, all subsequent callers not subject to C-O-S Req.

Example: To Handle or Not To Handle?

```
class A {
  ma(int i) {
    if(i < 0) { /* Error */ }
    else { /* Do something. */ }
  } }
```

```
class B {
  mb(int i) {
    A oa = new A();
    oa.ma(i); /* Error occurs if i < 0 */
  } }
```

```
class Tester {
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    int i = input.nextInt();
    B ob = new B();
    ob.mb(i); /* Where can the error be handled? */
  } }
```

```
class NegValException extends Exception {
  NegValException(String s) { super(s); }
}
```

context	caller	callee
A	/	/
B	B.mb	A.ma
Tester	Tester.main	B.mb

Version 1:
Handle it in `B.mb`

Version 2:
Pass it from `B.mb` and handle it in `Tester.main`

Version 3:
Pass it from `B.mb`, then from `Tester.main`, then throw it to the console.

call stack