Lecture 6 - Sep. 24

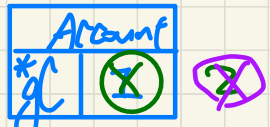
Review of OOP, Exceptions

Static Variables: Common Errors
Caller vs. Callee
Tracing Method Call Chain via Call Stack

Announcements/Reminders

- Lab1 released
- Mockup Programming Test this Fri (5pm or 6pm)
- Guides for WrittenTest1 and ProgTest1 to be released
- Reminder of rules for class attendance checks

Managing Account IDs: Automatic




```
class Account {
  private static int globalCounter = 1;
  private int id; String owner;
  public Account(String owner) {
    this.id = globalCounter;
    globalCounter++;
    this.owner = owner; } }
```

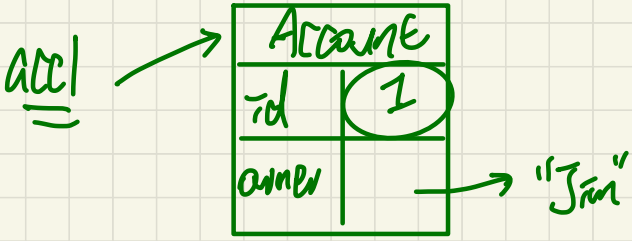
shared. init

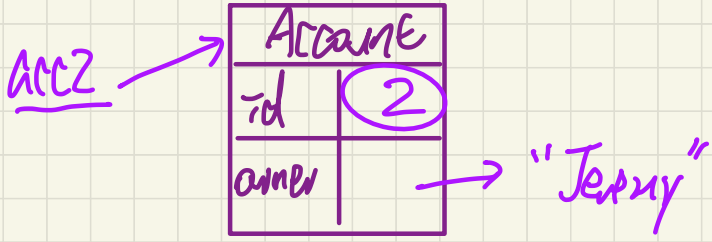
parameters not including id.

local — each obj of type Account has its own copy

Q. What if "id" is also static?

```
class AccountTester {
  Account acc1 = new Account("Jim");
  Account acc2 = new Account("Jeremy");
  System.out.println(acc1.getID() != acc2.getID()); }
```





Alt.

```
class Account {
    static int gC = 1;
    static int id;
    public Account() {
        id = gC;
        gC++;
    }
}
```

```
Account a1 = new Account();
Account a2 = new Account();
```

Account	
gC	~~1~~ ~~2~~ 3
id	~~1~~ ~~2~~

a2 → Account

a1 → Account

① a1.id 1

② a2.id 2

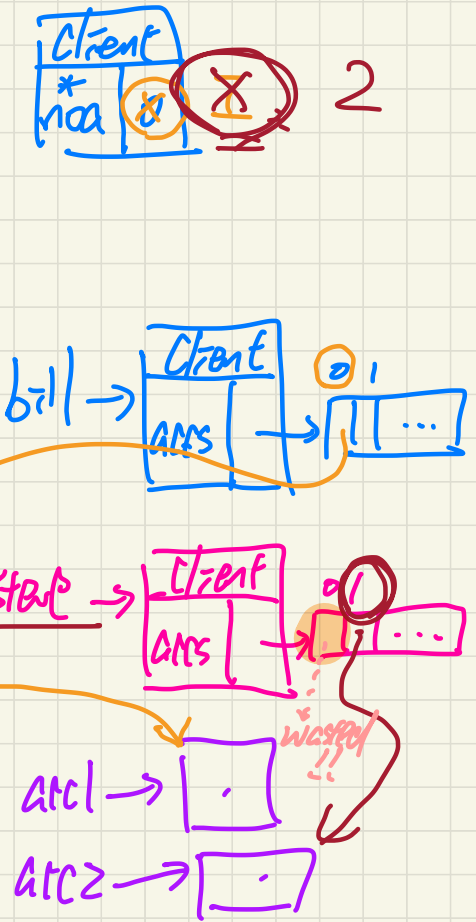
a1.id 2

↳ wrong!

Misuse of Static Variables


```
public class Client {
  private Account[] accounts;
  private static int numberOfAccounts = 0;
  public void addAccount(Account acc) {
    accounts[this.numberOfAccounts] = acc;
    this.numberOfAccounts ++;
  } }
```

```
public class ClientTester {
  Client bill = new Client("Bill");
  Client steve = new Client("Steve");
  Account acc1 = new Account();
  Account acc2 = new Account();
  bill.addAccount(acc1);
    /* [redacted] */
  steve.addAccount(acc2);
    /* [redacted] */
}
```



Use of Static Variables: Common Error

Slides 80 - 82

```
Bank cibc = new Bank();
cibc.branchName();
```

```
1  public class Bank {
2      private string branchName;
3      public String getBrachName() { return this.branchName; }
4      private static int nextAccountNumber = 0;
5      public static String getInfo() {
6          nextAccountNumber++;
7          return this.branchName + nextAccountNumber;
8      }
9  }
```

non-static

static

Bank.getInfo()

class name

static method

Compilation error!

To call this static method:

Bank.getInfo()

should be a C.O.

Bank.branchName()

non-static

Fix 1:

```
1  public class Bank {
2     private string branchName;
3     public String getBrachName() { return this.branchName; }
4     private static int nextAccountNumber = 0;
5     public static String getInfo() {
6        nextAccountNumber++;
7        return this.branchName + nextAccountNumber;
8     }
9  }
```

Fix 2:

```
1  public class Bank {
2     private string branchName;
3     public String getBrachName() { return this.branchName; }
4     private static int nextAccountNumber = 0;
5     public static String getInfo() {
6        nextAccountNumber++;
7        return this.branchName + nextAccountNumber;
8     }
9  }
```

static

Is there a shared branch name by all Banks? No!!

```
public class MyClass {
    public static void main ( ... ) {

    }
}
```

main

↓

Entry point of execution

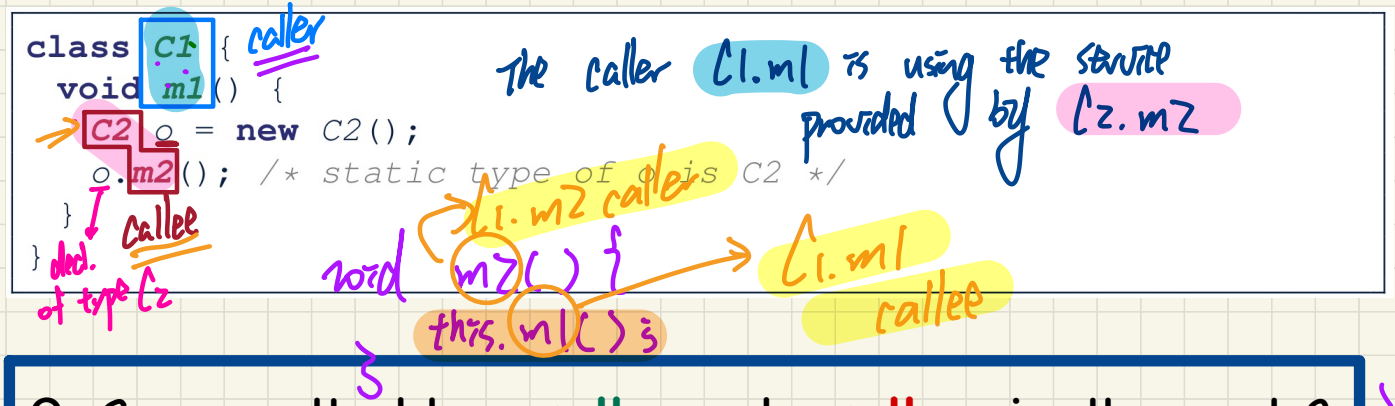
(MyClass.main())

↓

when exec. starts; no object creation is necessary

Caller vs. Callee

- **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? YES!

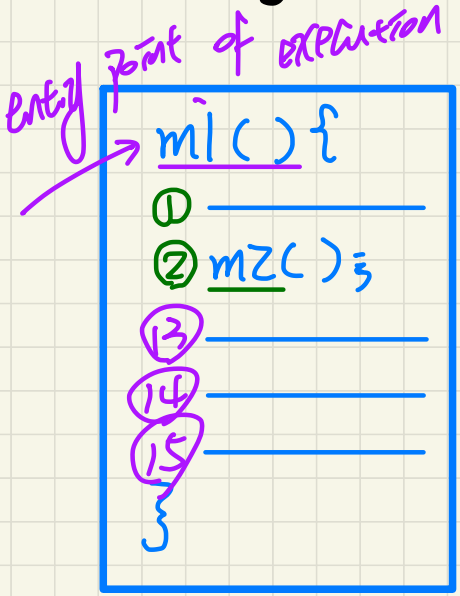
(1) Make C1.m1 a callee as well

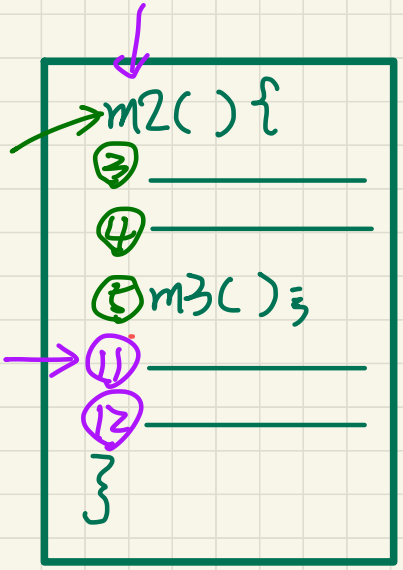
(2) Exercise. Make C2.m2 a caller.

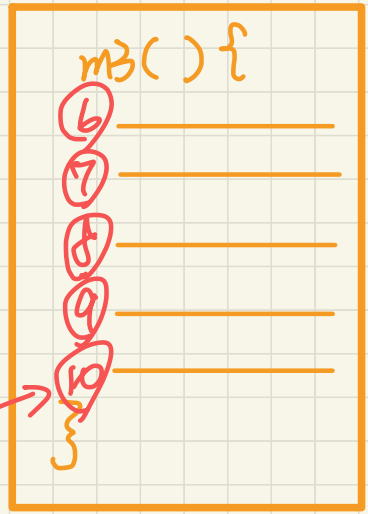
```
class C3 {
  void m3() {
    C1 o = new C1(); o.m1();
  }
}
```

C1.m1 is now a caller.

Visualizing a Call Chain using a Stack







push / add

pop / remove