Lecture 14 - Oct 28

Object Equality

Overridden equals: Phases 1, 2, 3
Static vs. Dynamic Types, Type Casting

Announcements/Reminders

- Today's class: notes template posted
- ProgTest0 and ProgTest1 marks and feedback released
- WrittenTest1 this Wednesday (October 29)
  + Some practice questions on eClass
  + Past review session materials released
  + Lecture 12 and Lecture 13 this week are covered.
- Lab3 released

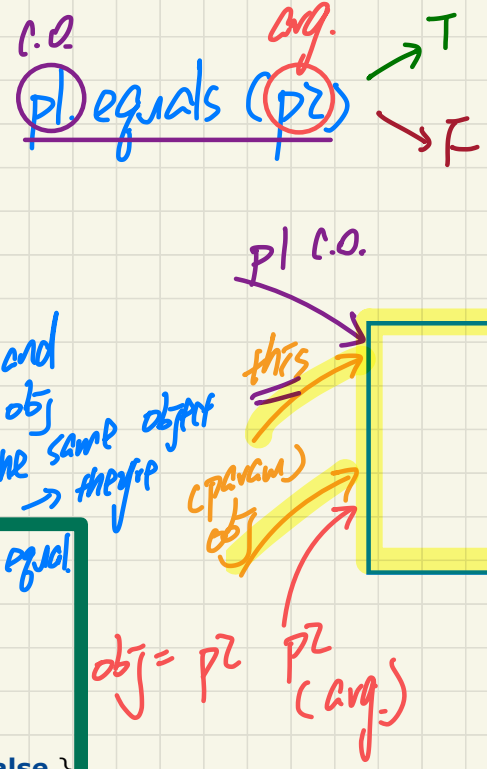
The equals Method: Overridden Version

Phase 1

```
public class Object {
   ...
   public boolean equals(Object obj) {
      return this == obj;
   }
}
```

extends

```
public class PointV2 {
  private int x;
  private int y;
  public PointV2 (int x, int y) { ... }
  public boolean equals(Object obj) {
    if(this == obj) { return true; }
    if(obj == null) { return false; }
    if(this.getClass() != obj.getClass()) { return false }
    PointV2 other = (PointV2) obj;
    return this.x == other.x
        && this.y == other.y;
  }
}
```



The equals Method: Overridden Version

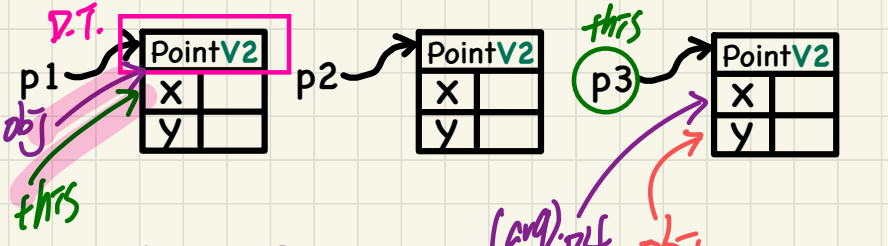
Example 1: Trace L7

```
public class Object {
   ...
   public boolean equals(Object obj) {
      return this == obj;
   }
}
```

extends

```
public class PointV2 {
  private int x;
  private int y;
  public PointV2 (int x, int y) { ... }
  public boolean equals(Object obj) {
    if(this == obj) { return true; }
    if(obj == null) { return false; }
    if(this.getClass() != obj.getClass()) { return false }
    PointV2 other = (PointV2) obj;
    return this.x == other.x
        && this.y == other.y;
  }
}
```

```
1  String s = "(2, 3)";
2  PointV2 p1 = new PointV2(2, 3);
3  PointV2 p2 = new PointV2(2, 3);
4  PointV2 p3 = new PointV2(4, 6);
5  System.out.println(p1 == p2);   /* [redacted] */
6  System.out.println(p2 == p3);   /* [redacted] */
7  System.out.println(p1.equals(p1));   /* [redacted] */
8  System.out.println(p1.equals(null));   /* [redacted] */
9  System.out.println(p1.equals(s));   /* [redacted] */
10 System.out.println(p1.equals(p2));   /* [redacted] */
11 System.out.println(p2.equals(p3));   /* [redacted] */
```



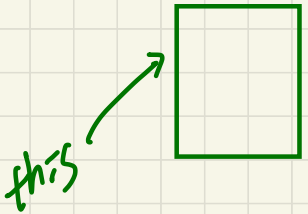
The equals Method: Overridden Version

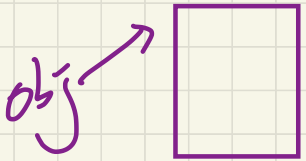
Phase 2

p1.equals(p2)

this == null → NPE.

* alt. phase 2






```
public class Object {
   ...
   public boolean equals(Object obj) {
      return this == obj;
   }
}
```

extends

```
public class PointV2 {
  private int x;
  private int y;
  public PointV2 (int x, int y) { ... }
  public boolean equals(Object obj) {
    if(this == obj) { return true; }
    if(obj == null) { return false; }
    if(this.getClass() != obj.getClass()) { return false }
    PointV2 other = (PointV2) obj;
    return this.x == other.x
        && this.y == other.y;
  }
}
```

reaching this line means: this != obj

unreachable :(if this were null, then a NPE would have already occurred.

```
if(obj == null) {
  if(this == null){
    return true;
  }else{ // this != null
    return false;
  }
}
```

PointV2	
x	
y	

The equals Method: Overridden Version

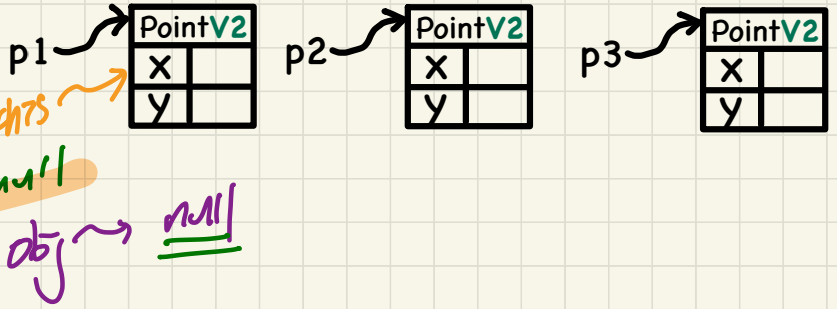
Example 1: Trace L8

```
public class Object {
   ...
   public boolean equals(Object obj) {
      return this == obj;
   }
}
```

extends

```
public class PointV2 {
  private int x;
  private int y;
  public PointV2 (int x, int y) { ... }
  public boolean equals(Object obj) {
    if(this == obj) { return true; }
    if(obj == null) { return false; }
    if(this.getClass() != obj.getClass()) { return false }
    PointV2 other = (PointV2) obj;
    return this.x == other.x
        && this.y == other.y;
  }
}
```

```
1  String s = "(2, 3)";
2  PointV2 p1 = new PointV2(2, 3);
3  PointV2 p2 = new PointV2(2, 3);
4  PointV2 p3 = new PointV2(4, 6);
5  System.out.println(p1 == p2);   /* [redacted] */
6  System.out.println(p2 == p3);   /* [redacted] */
7  System.out.println(p1.equals(p1));   /* [redacted] */
8  System.out.println(p1.equals(null));   /* [redacted] */
9  System.out.println(p1.equals(s));   /* [redacted] */
10 System.out.println(p1.equals(p2));   /* [redacted] */
11 System.out.println(p2.equals(p3));   /* [redacted] */
```



Static Type, Dynamic Type, Type Casting

Static Type: a ref variable's **declared** type

Dynamic Type: type of address **currently** stored in a ref variable

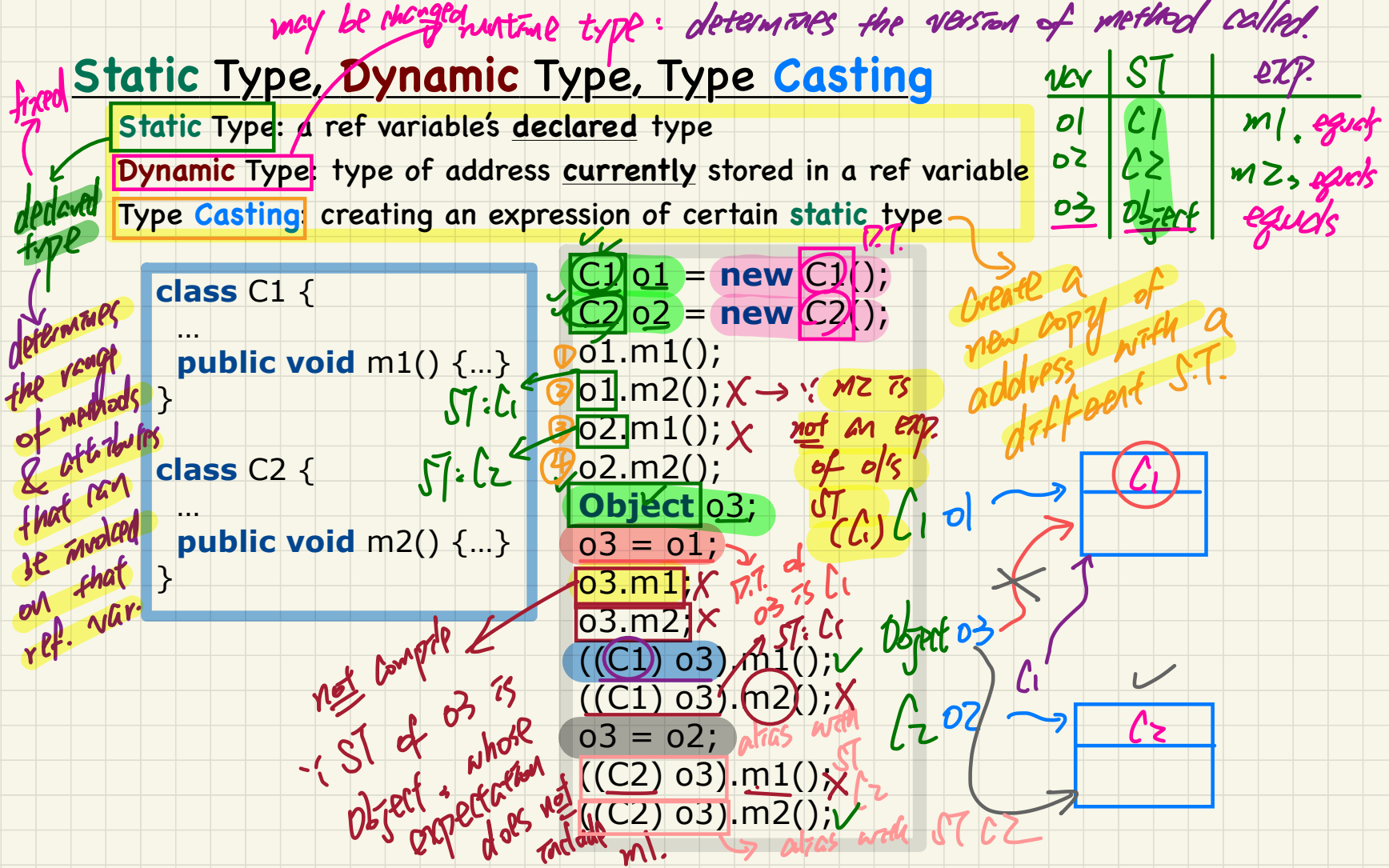
Type Casting: creating an expression of certain **static** type

var	ST	exp.
o1	C1	m1, equals
o2	C2	m2, equals
o3	Object	equals

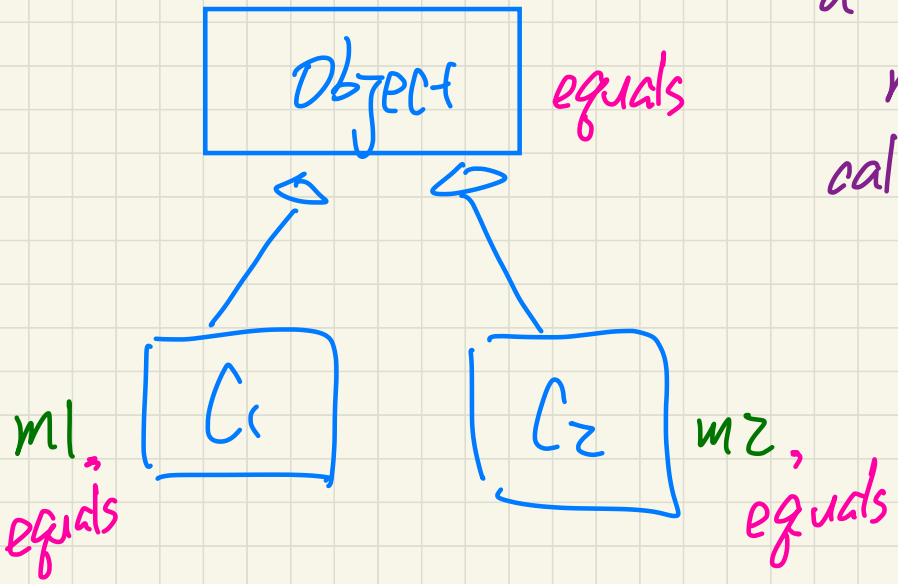
```
class C1 {
  ...
  public void m1() {...}
}

class C2 {
  ...
  public void m2() {...}
}
```

```
C1 o1 = new C1();
C2 o2 = new C2();
o1.m1();
o1.m2();   X
o2.m1();   X
o2.m2();
Object o3;
o3 = o1;
o3.m1();   X
o3.m2();   X
((C1) o3).m1();   ✓
((C1) o3).m2();   X
o3 = o2;
((C2) o3).m1();   X
((C2) o3).m2();   ✓
```



Expectation of a type/class: range of meth./attrs callable on that type



Given a line of method call:

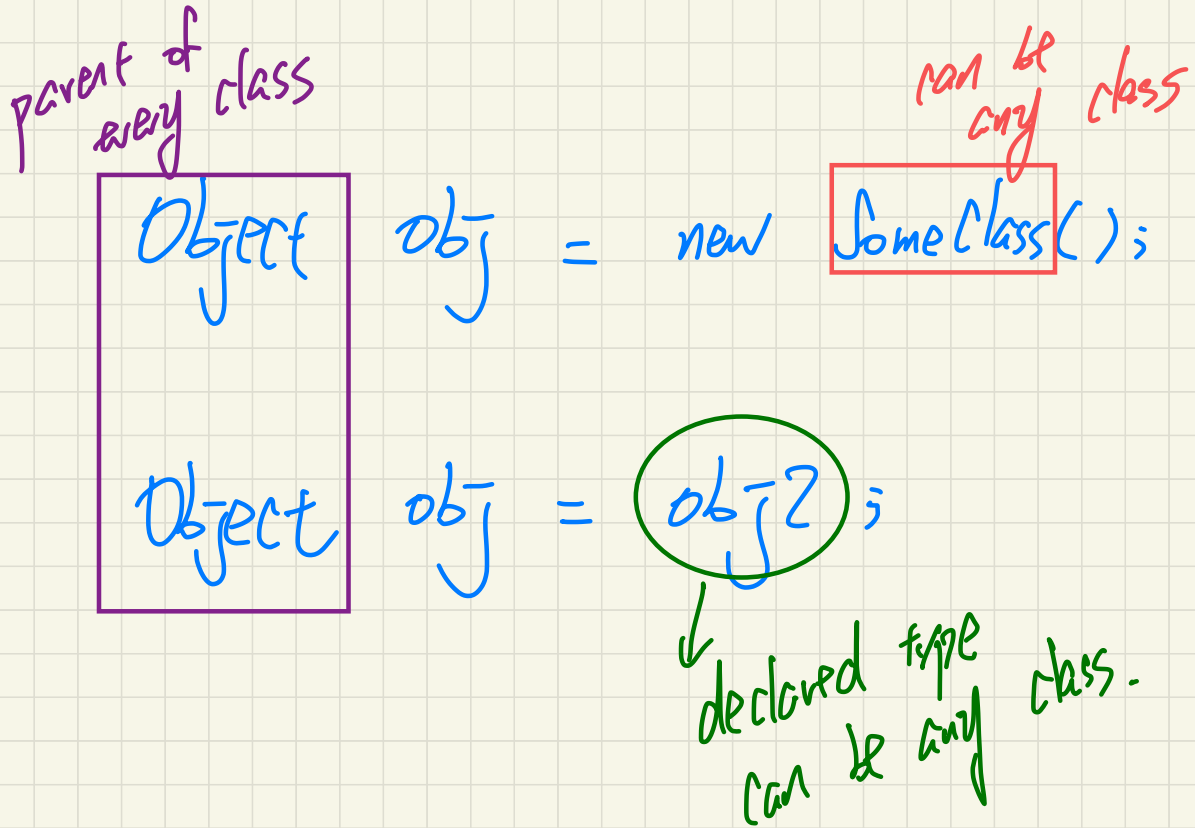
obj.m(...)

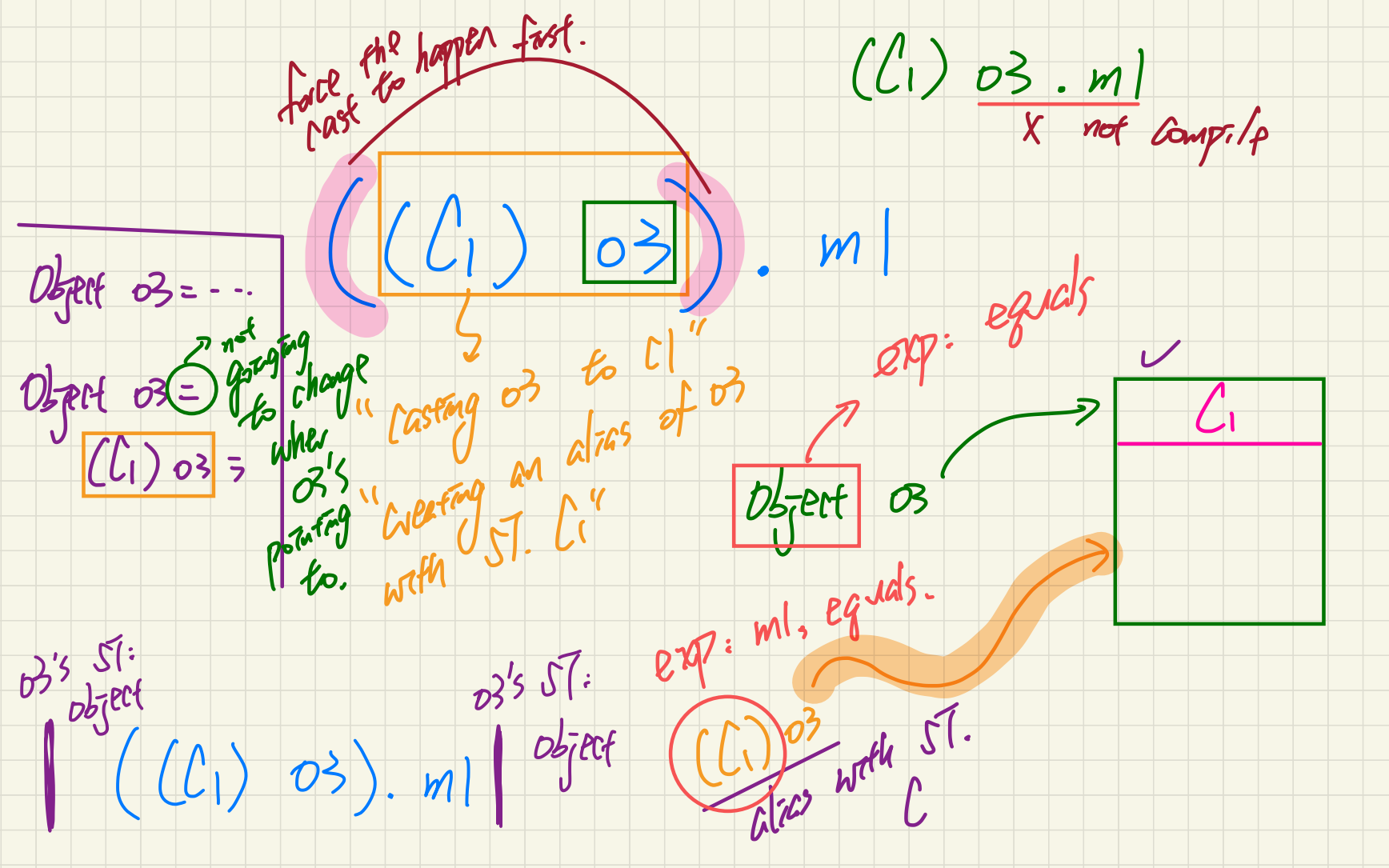
(1) Does it compile?

↳ look at the S.T. of C.O. obj.

(2) If it compiles, which version of m is called?

↳ look at the D.T of C.O. obj.


parent of every class
can be any class
Object obj = new SomeClass();
Object obj = obj2;
declared type can be any class.


force the cast to happen first.
(C1) o3 . m1
X not compile
((C1) o3) . m1
Object o3 = ...
Object o3 = o3
not going to change when o3's pointing to.
(C1) o3 =
"casting o3 to C1"
"creating an alias of o3 with ST. C1"
exp: equals
Object o3
C1
o3's ST: object
((C1) o3). m1
o3's ST: object
exp: m1, equals.
(C1) o3
alias with ST. C

The equals Method: Overridden Version

Phase 3

```
public class Object {
   ...
   public boolean equals(Object obj) {
      return this == obj;
   }
}
```

extends

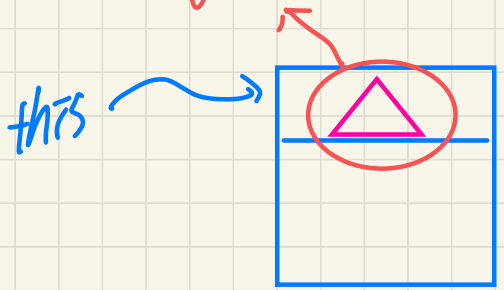
```
public class PointV2 {
  private int x;
  private int y;
  public PointV2 (int x, int y) { ... }
  public boolean equals(Object obj) {
    if(this == obj) { return true; }
    if(obj == null) { return false; }
    if(this.getClass() != obj.getClass()) { return false }
    PointV2 other = (PointV2) obj;
    return this.x == other.x
        && this.y == other.y;
  }
}
```

this != obj
obj != null

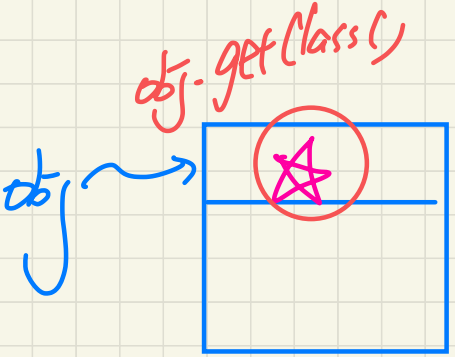
return the D.T. of the current objects

objects with different D.T.s are not equal to each other.

this.getClass()



obj.getClass()



The equals Method: Overridden Version

Example 1: Trace L9

```
public class Object {
   ...
   public boolean equals(Object obj) {
      return this == obj;
   }
}
```

extends

```
public class PointV2 {
  private int x;
  private int y;
  public PointV2 (int x, int y) { ... }
  public boolean equals(Object obj) {
    if(this == obj) { return true; }
    if(obj == null) { return false; }
    if(this.getClass() != obj.getClass()) { return false }
    PointV2 other = (PointV2) obj;
    return this.x == other.x
        && this.y == other.y;
  }
}
```

```
1  String s = "(2, 3)";
2  PointV2 p1 = new PointV2(2, 3);
3  PointV2 p2 = new PointV2(2, 3);
4  PointV2 p3 = new PointV2(4, 6);
5  System.out.println(p1 == p2);   /* [redacted] */
6  System.out.println(p2 == p3);   /* [redacted] */
7  System.out.println(p1.equals(p1));   /* [redacted] */
8  System.out.println(p1.equals(null));   /* [redacted] */
9  System.out.println(p1.equals(s));   /* [redacted] */
10 System.out.println(p1.equals(p2));   /* [redacted] */
11 System.out.println(p2.equals(p3));   /* [redacted] */
```

rm. false.