

EECS2030 Advanced Object-Oriented Programming
(Fall 2021)

Q&A - Lecture 5b

Wednesday, November 10

Announcement

1. polymorphic method param.
2. polymorphic method return types

- Lecture W9 (released: Nov. 2)
- **Lab4** released on Nov 8
- Written Test 2 & Programming Test 2

Inheritance
↳ Sendent
RS NRS

Incorrect Design of Test

Refer to the specific feedback on eClass.

Correct Design of Test

1. Confirming that everything works as expected.

2. If a method under test does not work as expected:

→ ① An error signaled

→ ② Exec. flow disrupted - without going further.

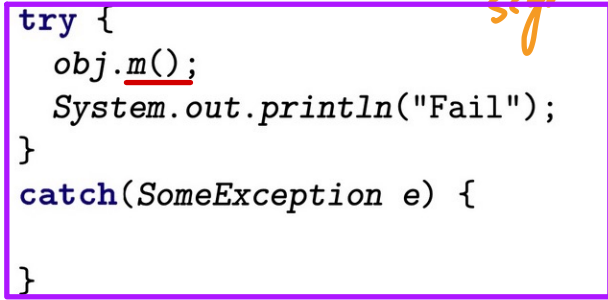
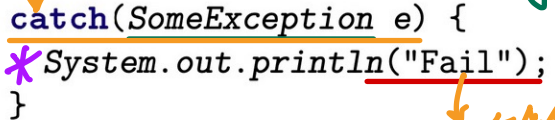
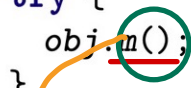
```
public class Tester { /* This is a console tester. */
    public static void main(String[] args) {
        SomeClass obj = new SomeClass();
        try {
            obj.m();
        }
        catch (SomeException e) {
            *System.out.println("Fail");
        }

        try {
            obj.m();
            System.out.println("Fail");
        }
        catch (SomeException e) {
        }
    }
}
```

→ first call to m expecting: no exception

↓ signaling error.

still executed even if an unexpected exception occurred at *.



PointV1 obj1 = new PointV1(3,4);

PointV2 obj2 = new PointV2(3,4);

assertTrue(obj1 == obj2); required that both operands are declared with the same type

↳ not compiling

assertSame(obj1, obj2); the same type

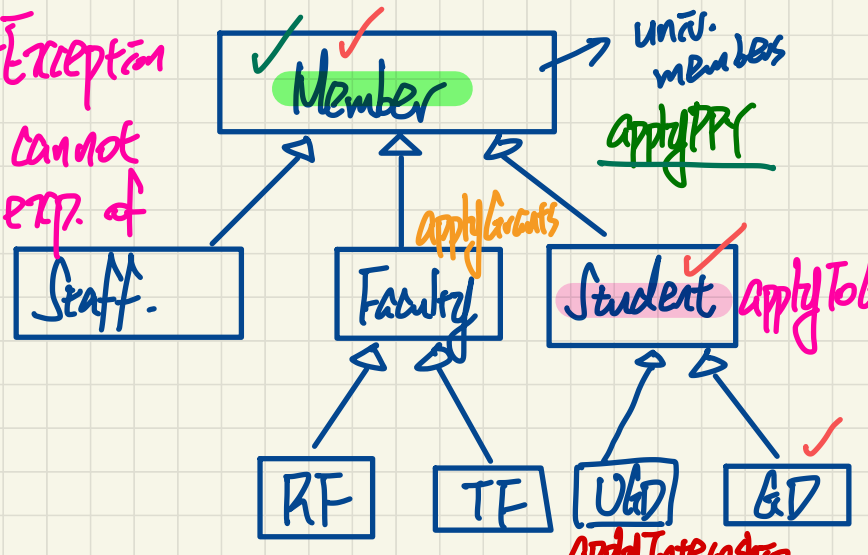
assertSame(Object o1, Object o2);

~~compiles but fails~~

method param.

refer to WF's lecture on polymorphic

b. ClassCastException
 ∴ DT S cannot fulfill exp. of
 Cast type
 GD.



Q1. Which ones
 compile?
 ↳ static
 type

- a. (Student) akn
 C.O. ST: Student
 applyToGrad.
- b. (GD) akn
 C.O. ST: GD
 applyToG

① Member akn = new Student(...);

- ② a. akn. applyPPY(c)
- ~~x~~ akn. applyToGraduate(c)
- ~~x~~ akn. applyGrants(c)

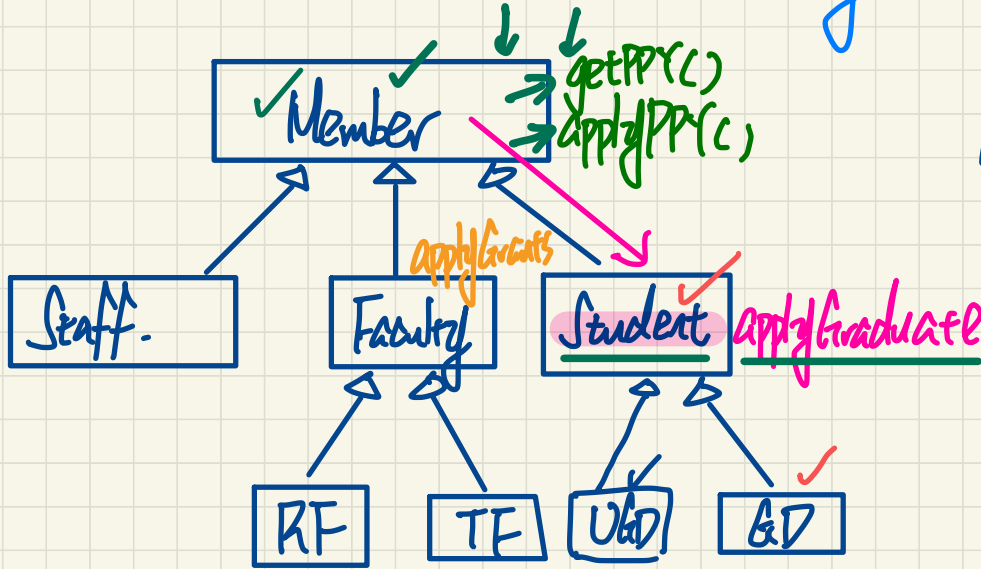
~~x~~ akn. applyInternship(c)

→ Q2. Which ones
 compile?

Q3. Runtime?
 a. ok ∴ DT S fulfills exp.

- ① casts compile
- ∴ downward. applyToG
- ② ST of cast exp. support

Access Control for Upward Casting



Upward cast to restrict
access to Member methods
only.

```

Member Student jim = new UGS(...);
  
```

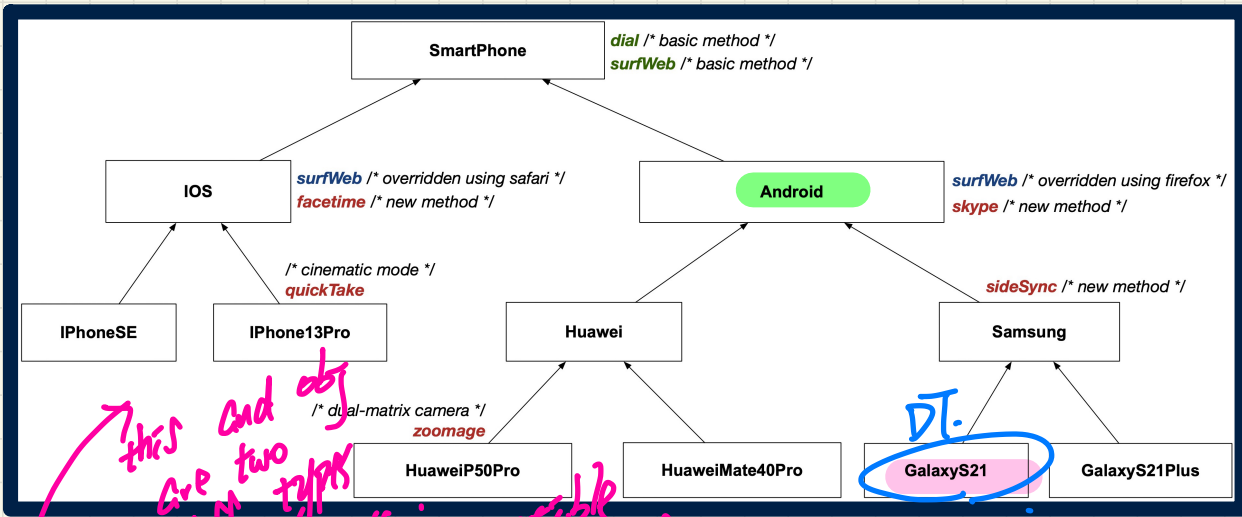
```

x po.applyForPass(Student jim);
po.applyToPass(Member jim);
  
```

```

class ParkingOffice {
    void applyForPass(Student Member m) {
        ...
        m.applyGraduate();
    }
}
  
```

→ inappropriate
: this not the
business of P.O.
to apply grad.



this and obj
 are two types
 diff. types
 of objects
 ⇒ incompatible
 to compare

if (this.getClass() instanceof Android) {
 obj.getClass() !=
 print ln
 return false;

myPhone = new GalaxyS21();
 (myPhone.getClass().getName())
 ↓
 retrieve DT of CO.
 ↓
 GalaxyS21

3