Lecture 19 - Nov 12

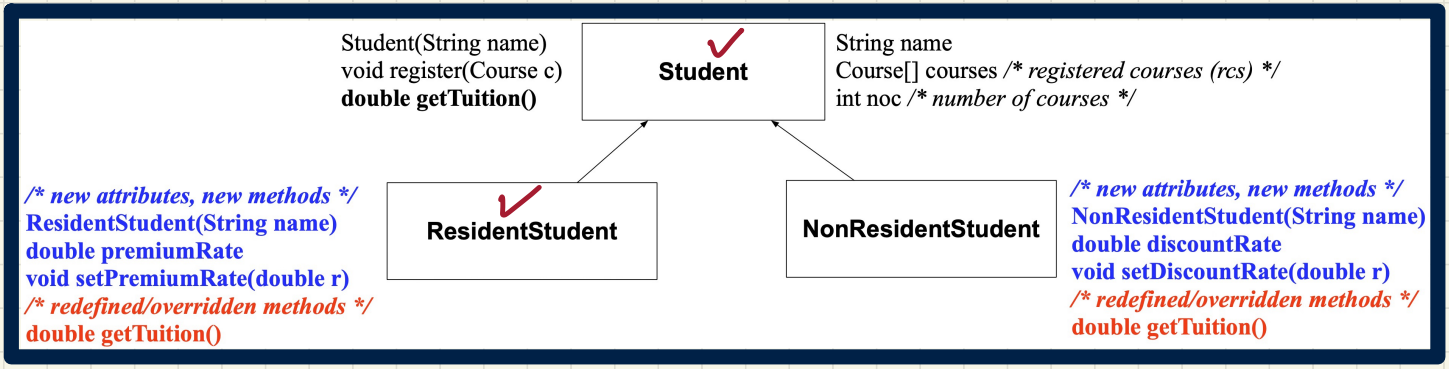
Inheritance

Type Casts: Motivation, Syntax, Visual
Compilable Casts: Upward vs. Downward
Casts at Runtime: ClassCastException

Announcements/Reminders

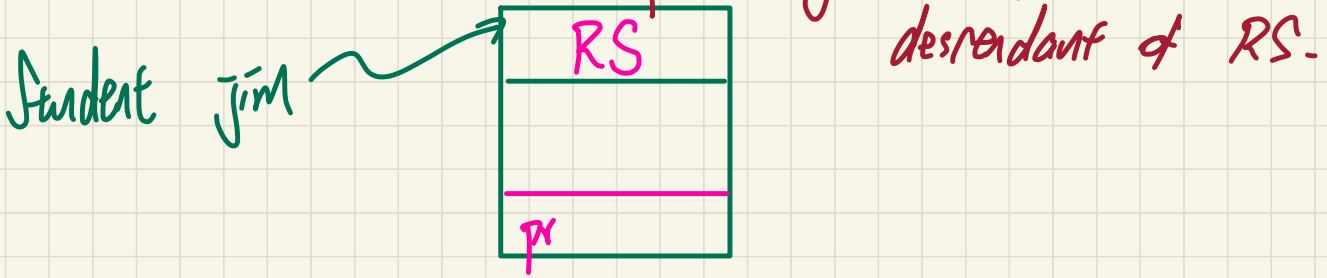
- Today's class: notes template posted
- Lab4 released
- WrittenTest2 guide and example questions released
- Notes on Type Casting

Type Cast: Motivation

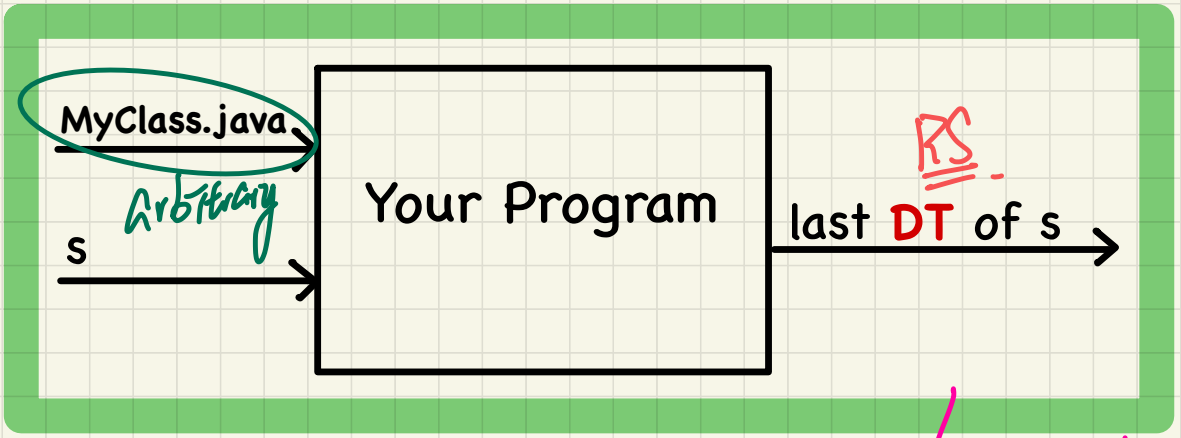



```
1  Student jim = new ResidentStudent("J. Davis");
2  ResidentStudent rs = jim;
3  rs.setPremiumRate(1.5);
```

✗ compile !! Jim's ST (S) not descendant of RS.



An A+ Challenge: Inferring the DT of a Variable



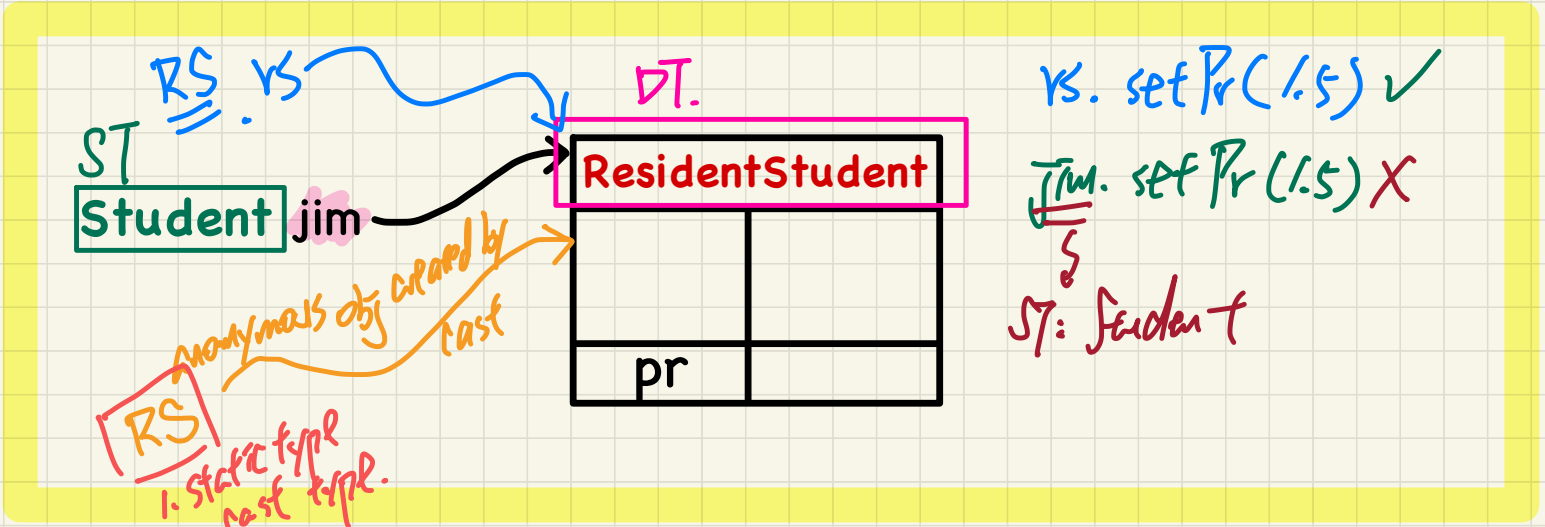

```
class MyClass {
  main (...)
    Student s = ...;
    ...
    s = new ResidentStudent(...);
  }
}
```

may not terminate

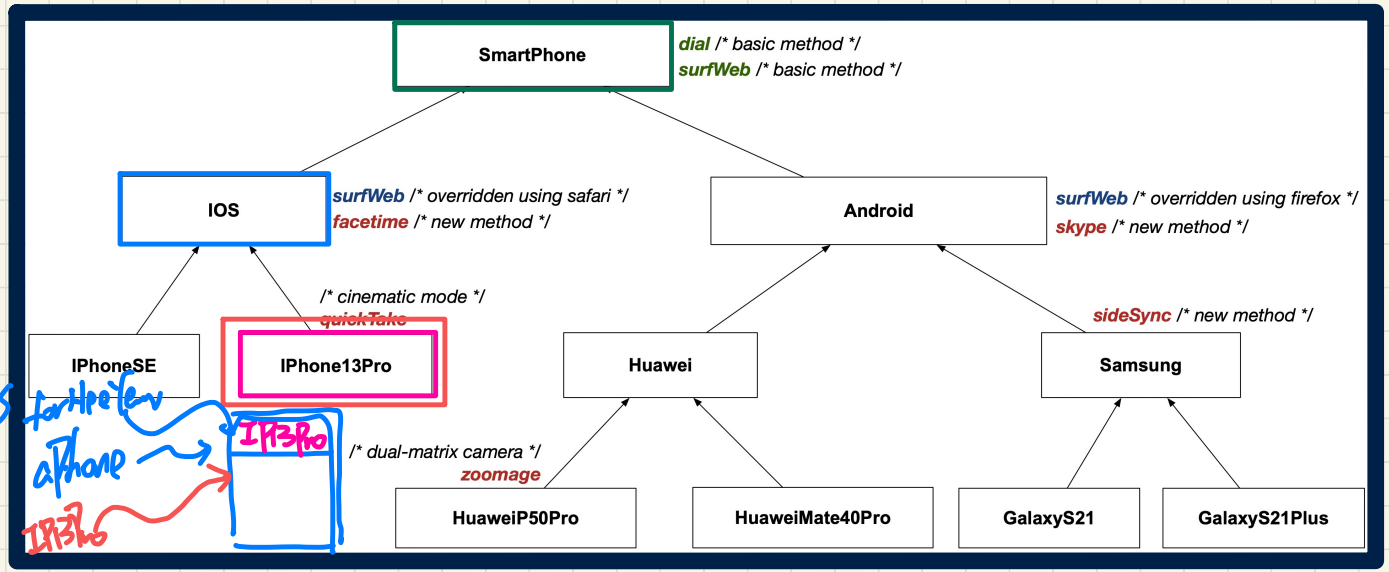
undecidable problem.
(not feasible to write a prog that infers the dynamic behaviour of another program)

Anatomy of a Type Cast

Student jim = **new** **ResidentStudent**("Jim");



Type Cast: Named vs. Anonymous



Named Cast: Use intermediate variable to store the cast result.

```
SmartPhone aPhone = new IPhone13Pro();
IOS forHeeyeon = (IPhone13Pro) aPhone;
forHeeyeon.facetime();
```

Anonymous Cast: Use the cast result directly.

```
SmartPhone aPhone = new IPhone13Pro();
(IPhone13Pro) aPhone).facetime();
```

Exercise

```
SmartPhone aPhone = new IPhone13Pro();
(IPhone13Pro) aPhone.facetime();
```

Compilable Casts: Upwards vs. Downwards

w.r.t. ST.

```
Android myPhone = new GalaxyS21Plus();

SmartPhone sp = (SmartPhone) myPhone;

GalaxyS21Plus ga = (GalaxyS21Plus) myPhone;
```

x to consider compilable cast, disregard DT.

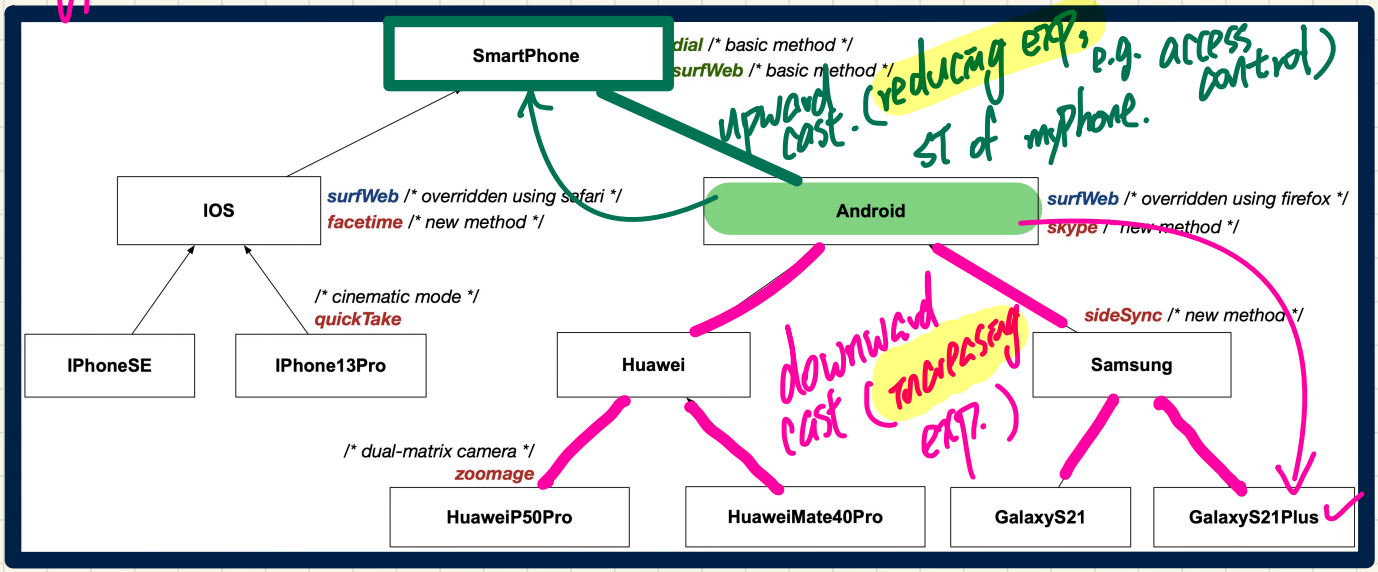
upward

downward

ST. skype X

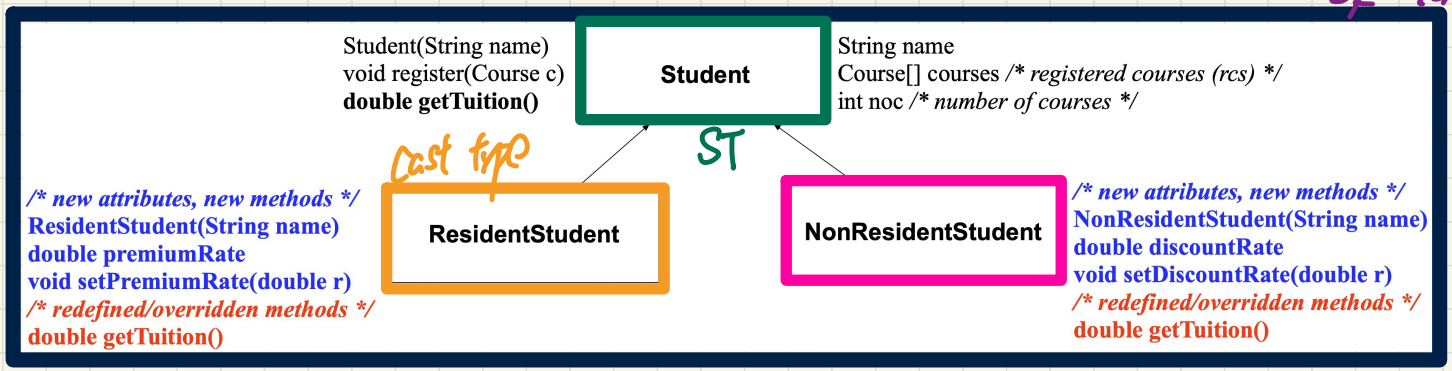
Expectations

	sp	myPhone	ga
dial	✓		
surfWeb	✓		
skype			
sideSync			
facetime			
quickTake			
zoomage			



Compilable Type Cast May Fail at Runtime (1)

(b) DT of jim (NRS) is not a descendant of cast type (RS)

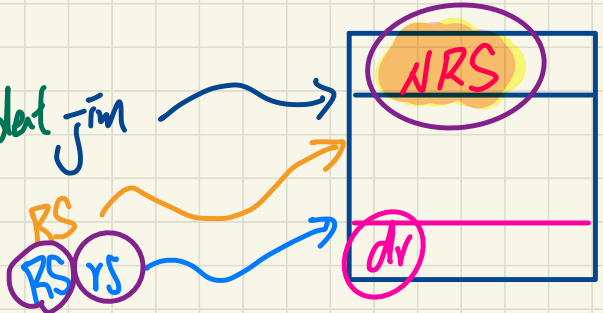



```
1  Student jim = new NonResidentStudent("J. Davis");
2  ResidentStudent rs = (ResidentStudent) jim;
3  rs.setPremiumRate(1.5);
```

Compiles ∵ downward cast

CCE

problematic ∵ DT NRS does not support student jim pr.

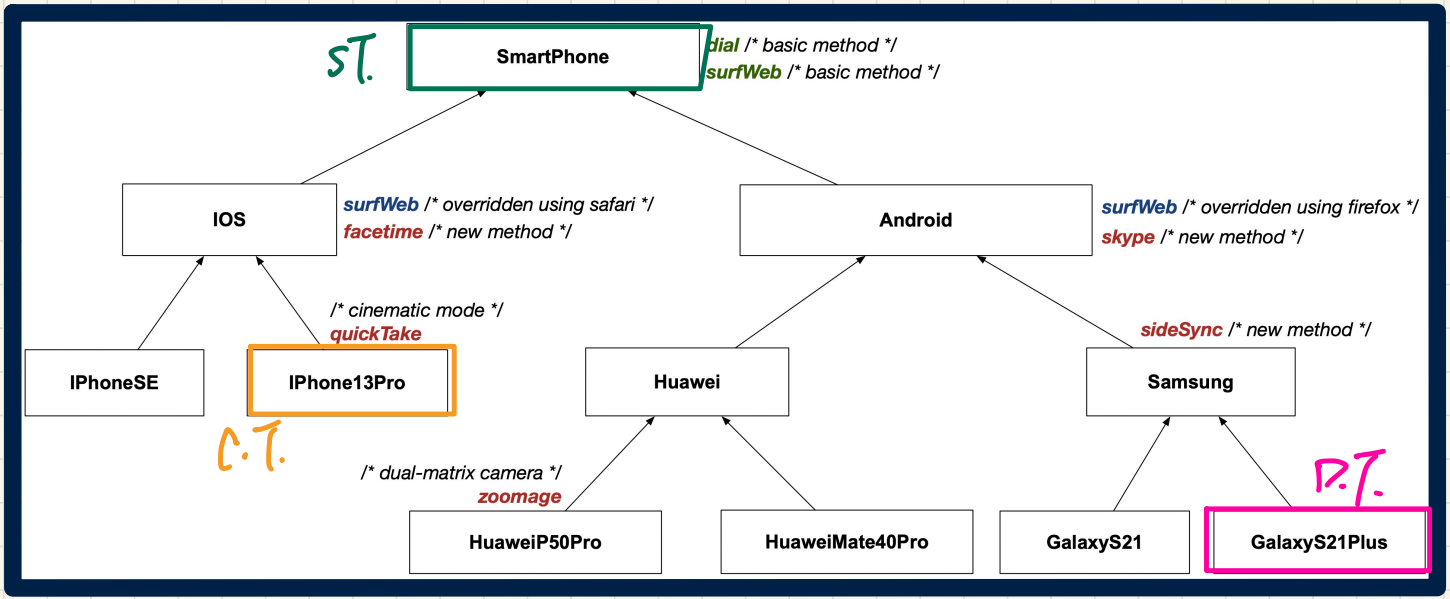


(1) compiles ∵ downward cast

(2) at runtime: ClassCastExcep.

(a) ∵ DT of jim (NRS) cannot fulfil exp. of cast type (RS)

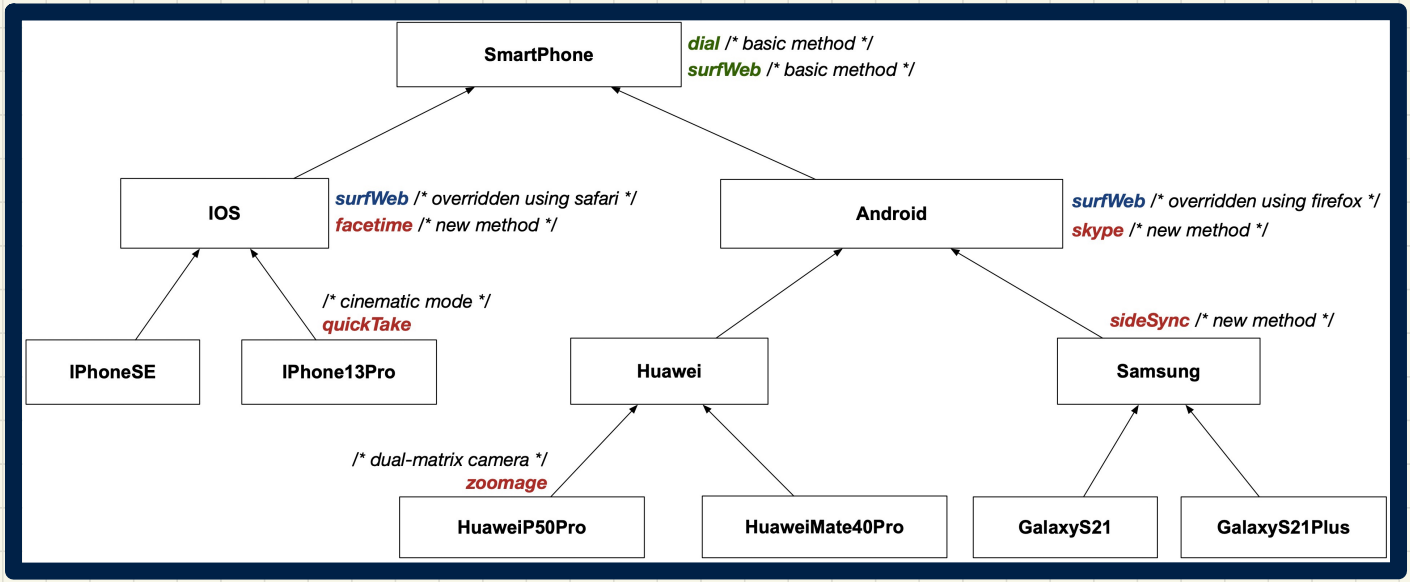
Compilable Type Cast May Fail at Runtime (2)




```
1  SmartPhone aPhone = new GalaxyS21Plus();
2  IPhone13Pro forHeeyeon = (IPhone13Pro) aPhone;
3  forHeeyeon.quickTake();
```

compiles but CCE (∵ D.T. GS21Plus not descendant of C.T. IP13P)

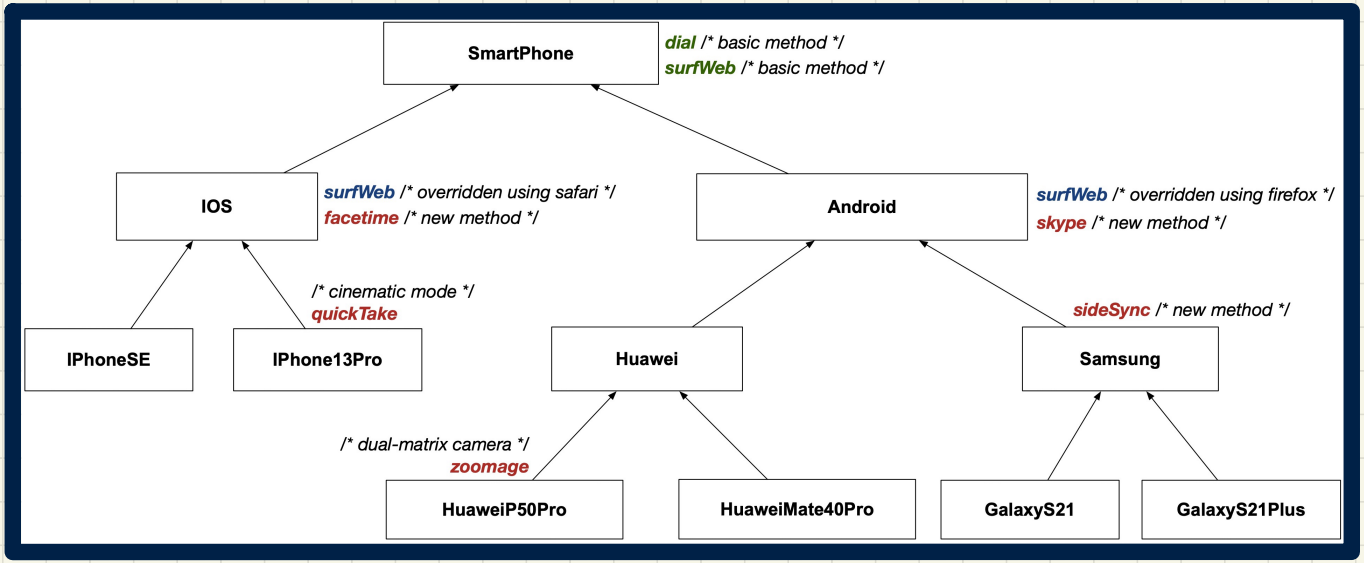
Exercise: Compilable Type Cast? Fail at Runtime? (1)




```
SmartPhone myPhone = new Samsung();
/* ST of myPhone is SmartPhone; DT of myPhone is Samsung */
GalaxyS21Plus ga = (GalaxyS21Plus) myPhone;
```

Compilable? ClassCastException at runtime?

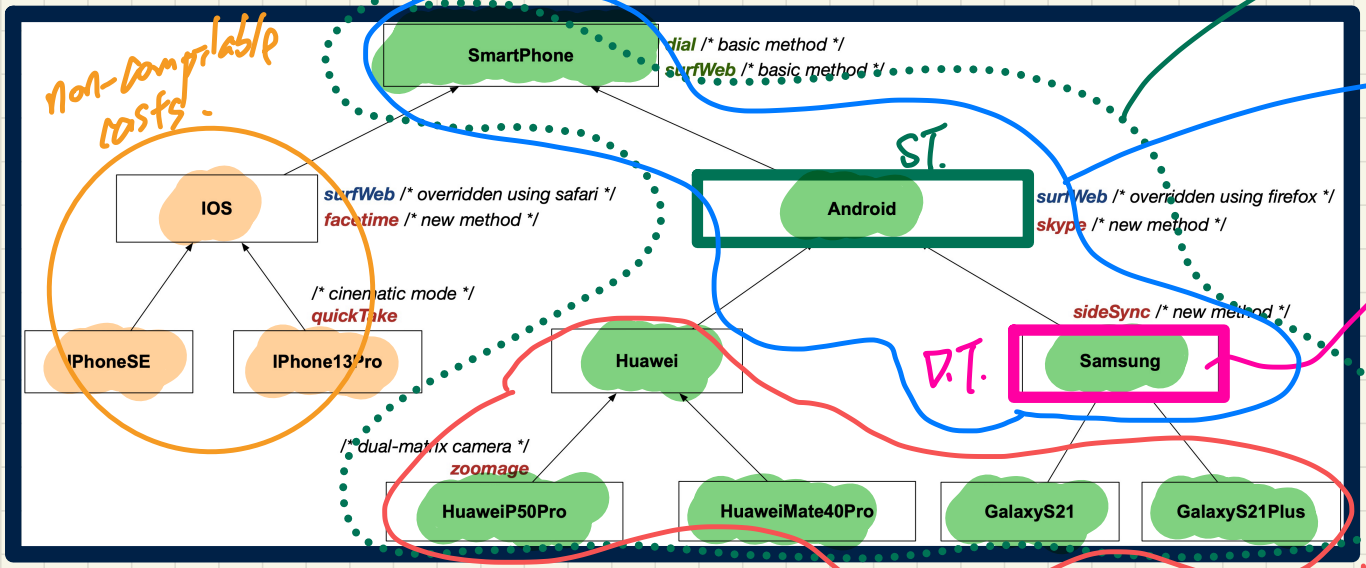
Exercise: Compilable Type Cast? Fail at Runtime? (2)




```
SmartPhone myPhone = new Samsung();
/* ST of myPhone is SmartPhone; DT of myPhone is Samsung */
IPhone13Pro ip = (IPhone13Pro) myPhone;
```

Compilable? ClassCastException at runtime?

Compilable Cast vs. Exception-Free Cast




```
Android myPhone = new Samsung();
```

Compilable Casts

Exception-Free Casts

Non-Compilable Casts

ClassCastException

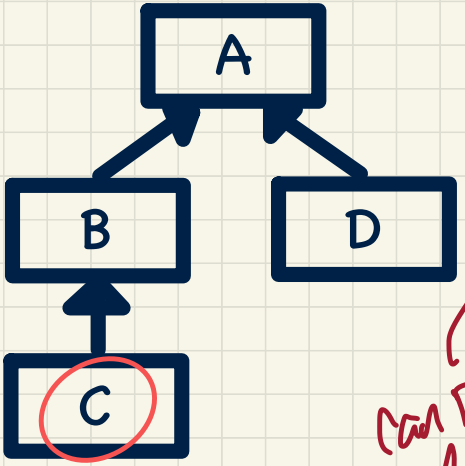
Exercise: Compilable Cast vs. Exception-Free Cast

```
class A { }
class B extends A { }
class C extends B { }
class D extends A { }
```

```
1  B b = new C();
2  D d = (D) b;
```

✓

✗ compile ∵ D is neither ancestor nor descendant of ST. of b (B)



D d = (D)((A) b)

upward cast

No. ∵ cast type D is not an ancestor of P.T. C.

↳ CCE!!

↳ compiles!

downward cast

can P.T. C fulfill D?

(D) ((A) b)

P.T.: C

can C fulfill exp(A)?