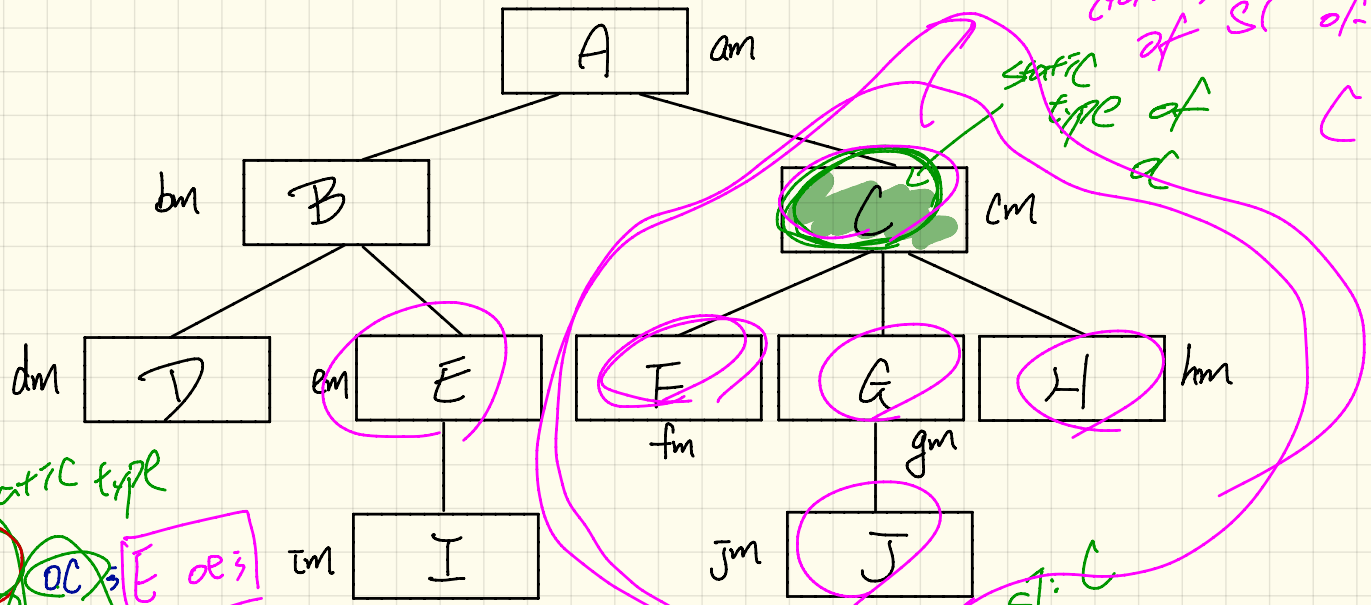
Wednesday Nov. 7

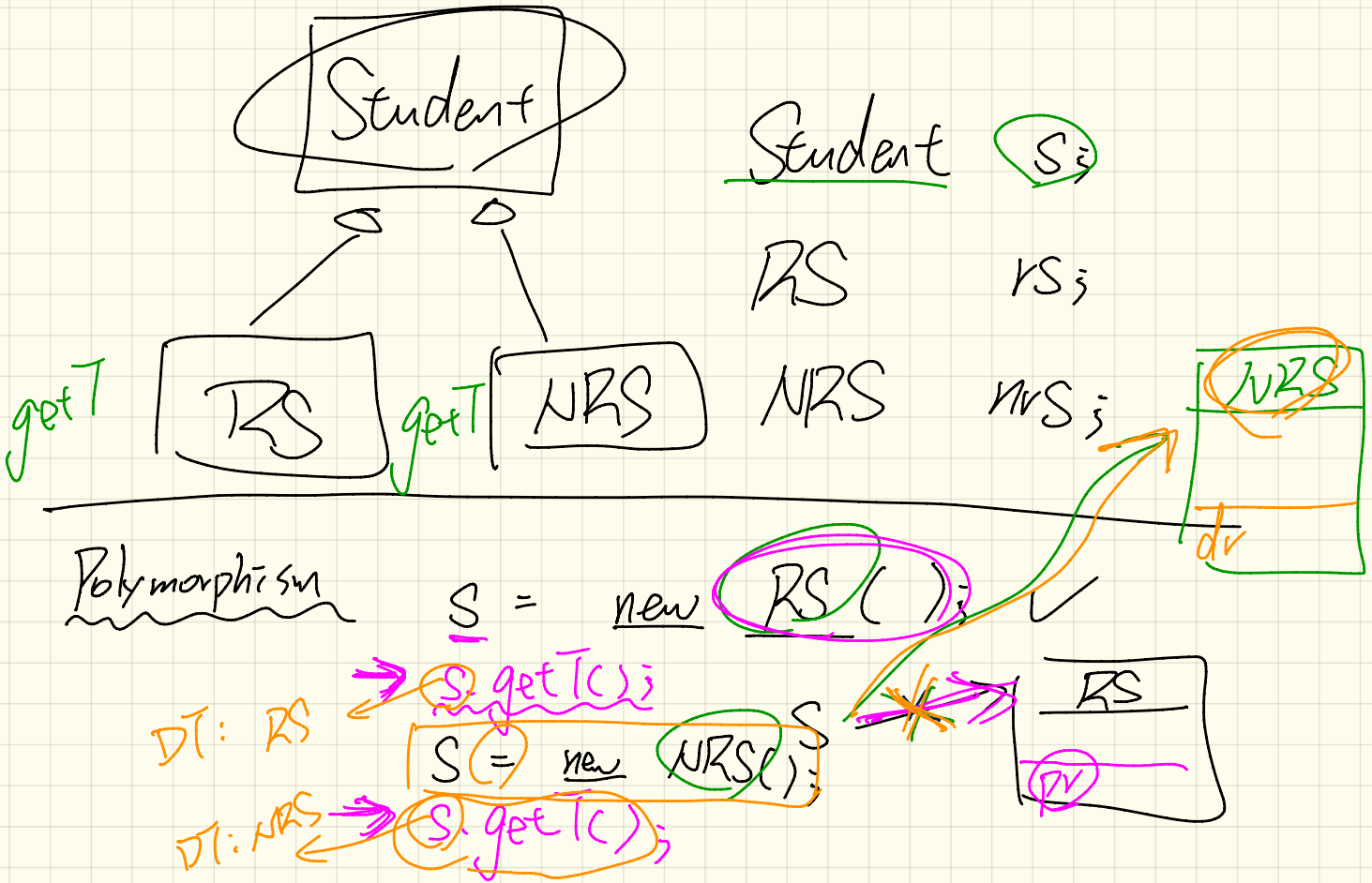
Lecture 17

Static Type vs. Dynamic Type

- Does the code compile? static type
- How does the compilable code behave at runtime? dynamic type

Rules of Substitutions




Student
getT
RS
getT
NRS
Student S;
RS rs;
NRS nrs;
nrs
dv
Polymorphism
S = new RS();
DT: RS
S.getT();
S = new NRS();
DT: NRS
S.getT();
RS
dv

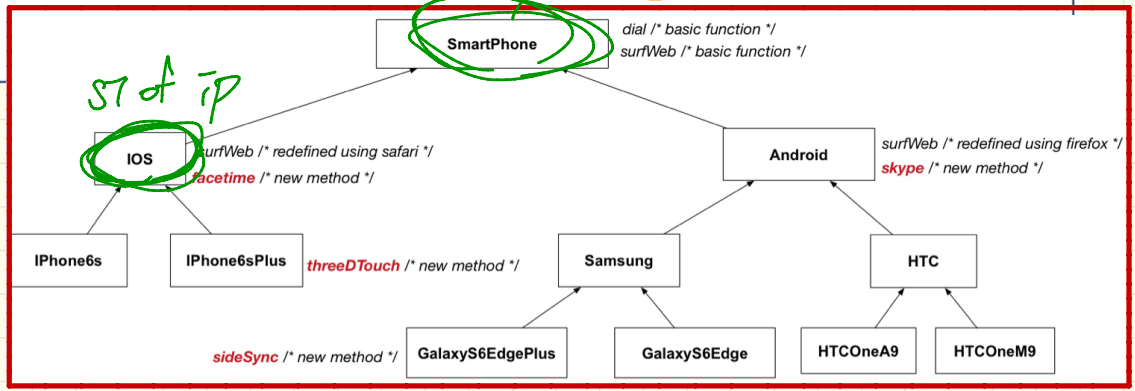
Polymorphism (2)

```
class SmartPhoneTest1 {
  public static void main(String[] args) {
    SmartPhone myPhone;
    IOS ip = new IPhone6sPlus();
    Samsung ss = new GalaxyS6Edge();
    myPhone = ip;   /* legal */
    myPhone = ss;   /* legal */

    IOS presentForHeeyeon;
    presentForHeeyeon = ip;   /* legal */
    presentForHeeyeon = ss;   /* illegal */
  }
}
```

ST: IOS

ST of myPhone



Dynamic Binding (2)

```
class SmartPhoneTest2 {
  public static void main(String[] args) {
    SmartPhone myPhone;
    IOS ip = new IPhone6sPlus();
    myPhone = ip;
    myPhone.surfWeb();  /* version of surfWeb in IPhone6sPlus */

    Samsung ss = new GalaxyS6Edge();
    myPhone = ss;
    myPhone.surfWeb();  /* version of surfWeb in GalaxyS6Edge */
  }
}
```

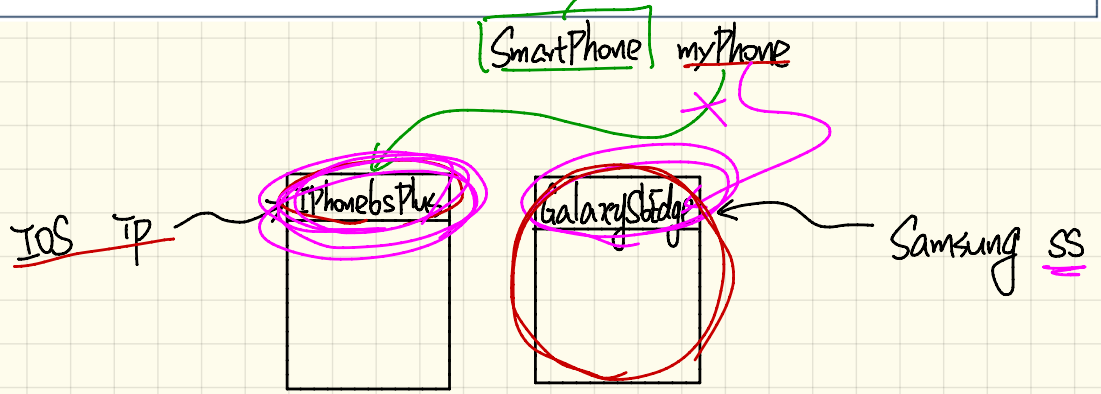
DT of ip is IP6sPlus

DT of myPhone: IPhone6sPlus

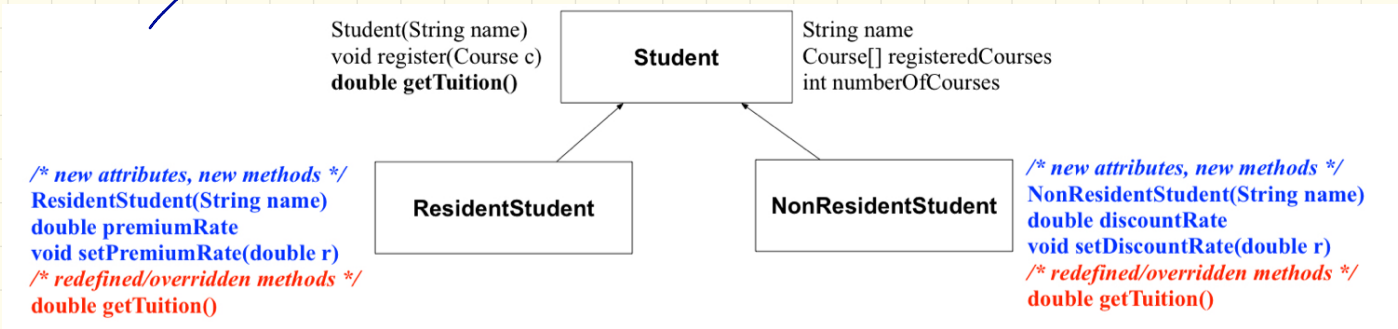
ST: SP

DT of myPhone: G6E

ST of myPhone

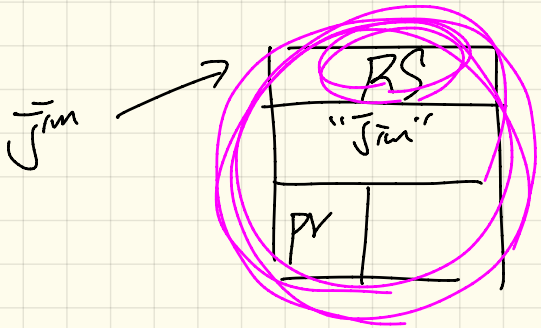


Type Cast: Motivation




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

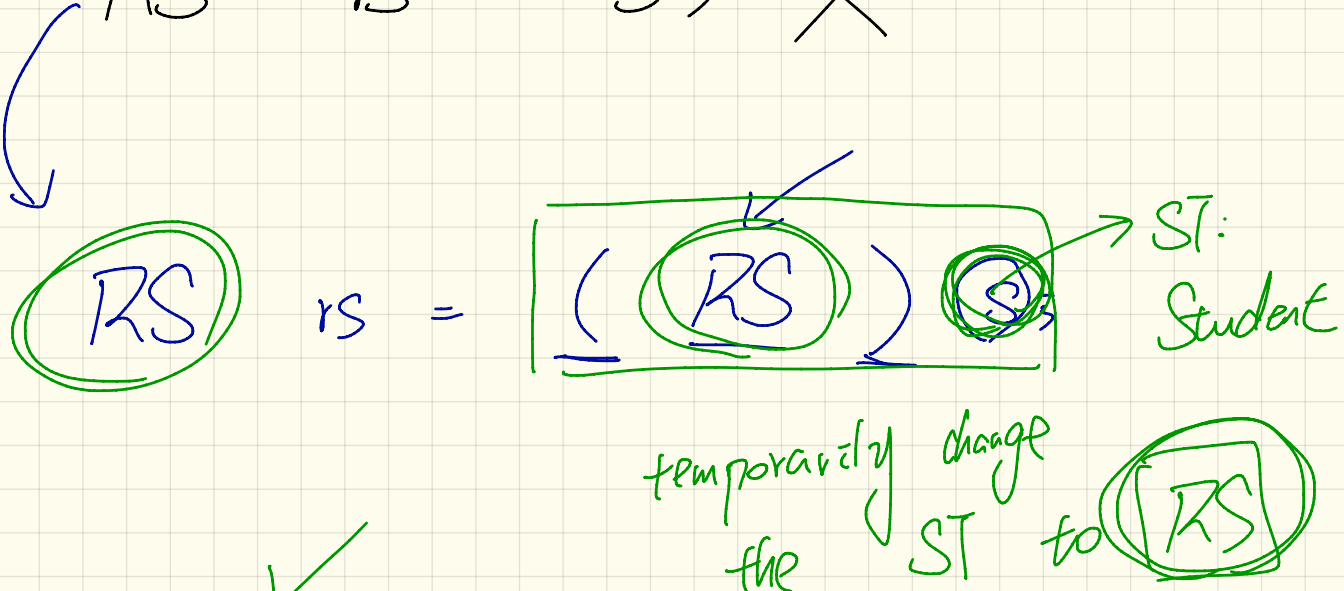
DT of jim? RS



At this point, jim's DT is really a RS, but Java compiler would not allow us to assign "jim" to a RS.

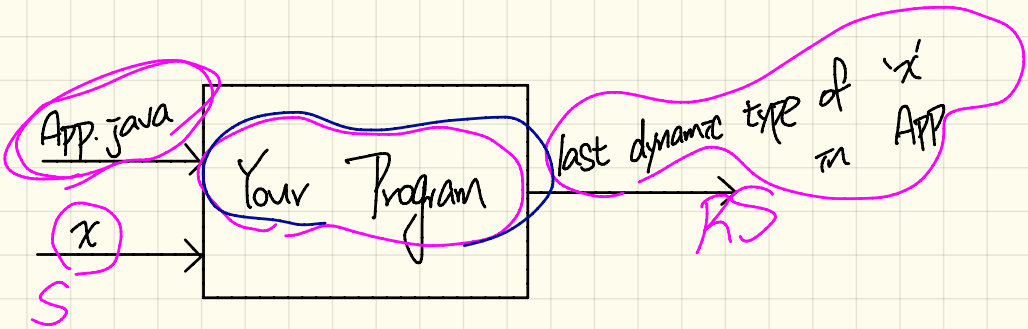
Student S = new RS(..);

RS rs = S; ✗



✓

Keeping Track of Dynamic Types: Undecidable



e.g.

App.java

```
Student S;
⋮
S = new RS(...);
```

```
Student S;
while (true) {

}
S = new RS(..);
```

Type Cast : Named or Anonymous

Named Cast

```
SmartPhone aPhone = new IPhone6sPlus();
IOS forHeeyeon = (IPhone6sPlus) aPhone;
forHeeyeon.facetime();
```

change the ST of aPhone to IP6sPlus

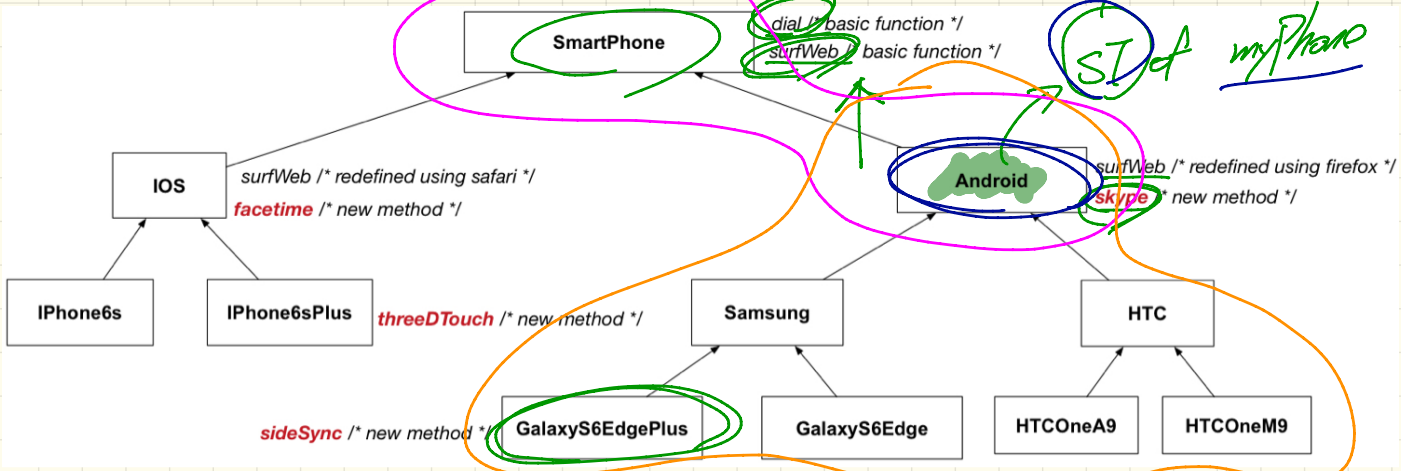
Anonymous Cast

```
SmartPhone aPhone = new IPhone6sPlus();
((IPhone6sPlus) aPhone).facetime();
```

Problem?

```
1  SmartPhone aPhone = new IPhone6sPlus();
2  (IPhone6sPlus) aPhone.facetime();
```

Compilable Cast : Upward vs. Downward




```
Android myPhone = new GalaxyS6EdgePlus();
SmartPhone sp = (SmartPhone) myPhone;
GalaxyS6EdgePlus ga = (GalaxyS6EdgePlus) myPhone;
```

Upward casting

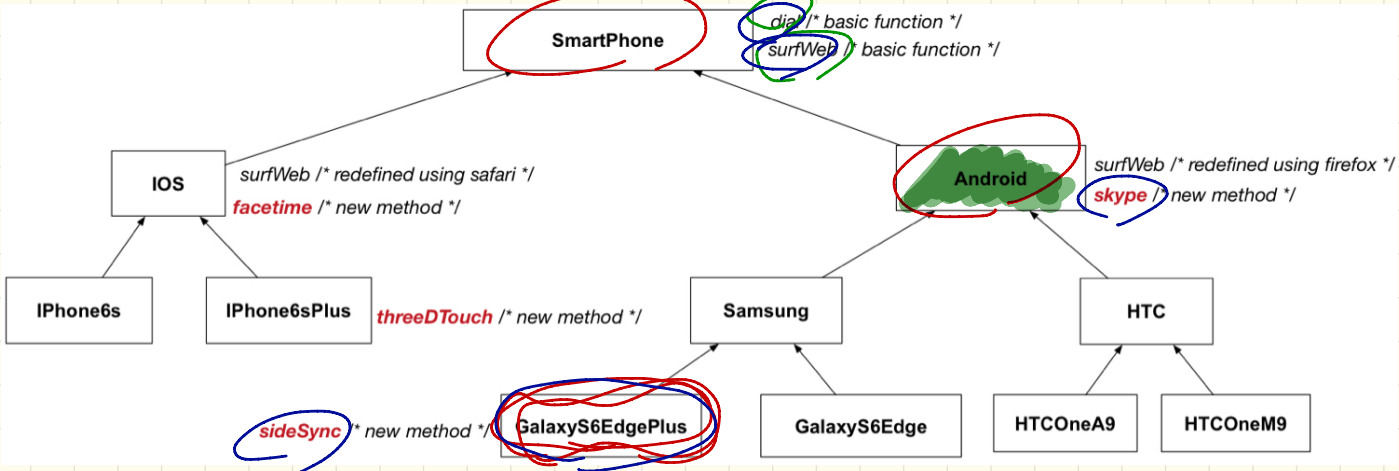
Downward casting

EXPECTATIONS

myPhone : [skype, surfWeb, dial]

sp : dial, surfWeb

ga : dial, surfWeb, skype, sideSync



Android

myPhone = - - - >

myPhone . dial surfweb skype

Upward casting

SmartPhone myPhone . dial surfweb

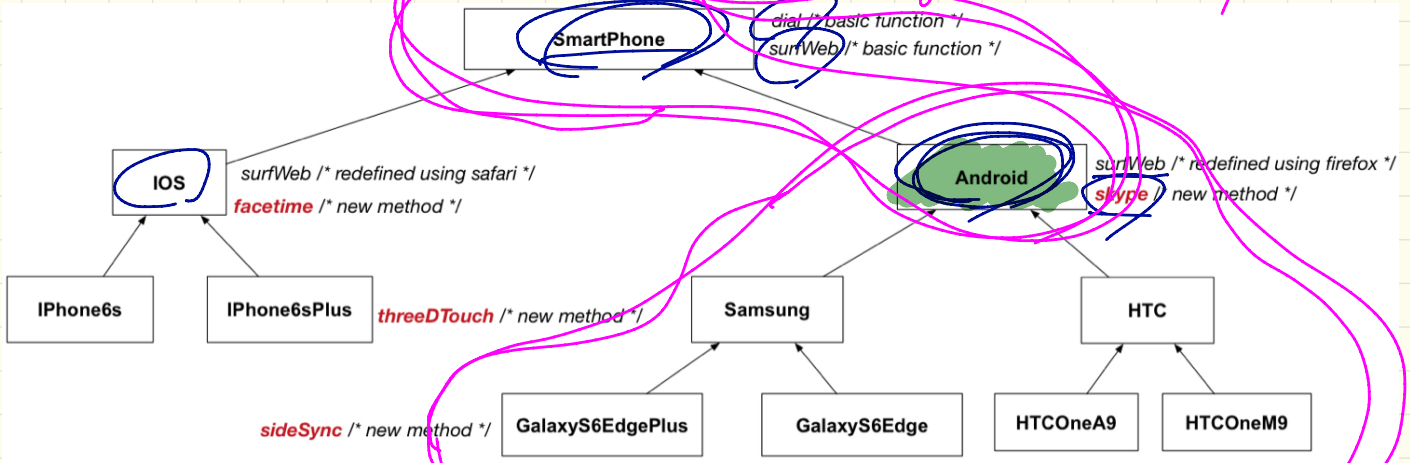
SmartPhone

Downward Casting

(GS6EP myPhone) . dial surfweb skype sideSync

ST: GS6EP

upward casting (narrow expectation)



Android p =

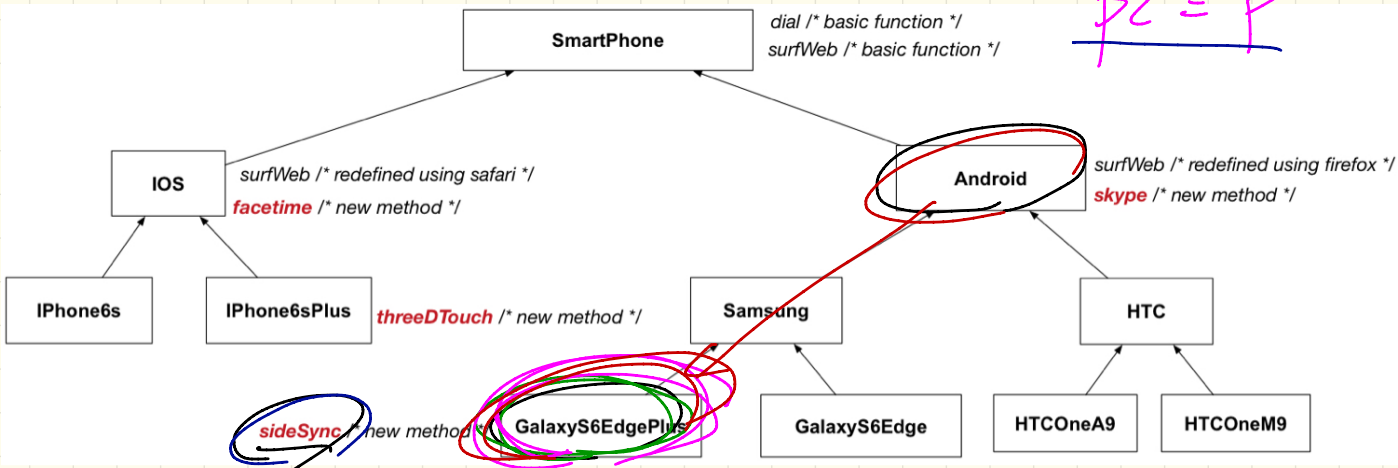
ST

p. skype
p. surfWeb
p. dial

downward casting (widen expectation)

What kind of (cast) will compile ?

↓ temporarily changes the ST.



p2 = p

Android p = new GS6EP(..);

p.skype
p.surfweb
p.dial

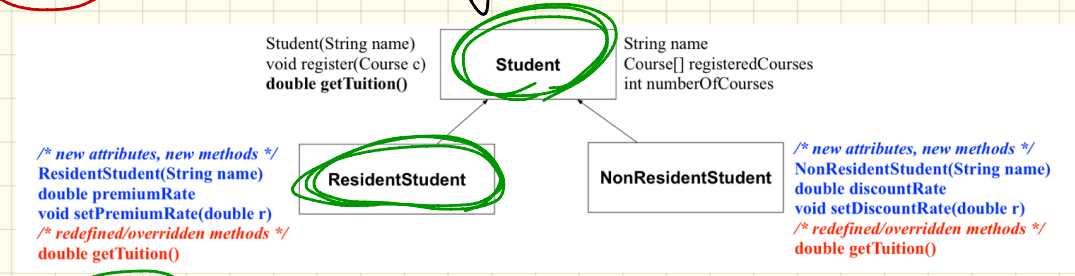
p.sideSync ✗
p.sideSync ✗

GS6EP p2 = GS6EP(..);
p2.sideSync ✓

Android p3 = (Android) p2;

GS6EP

Compilable Cast May Fail at Runtime (1)




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

downward casting → compile!