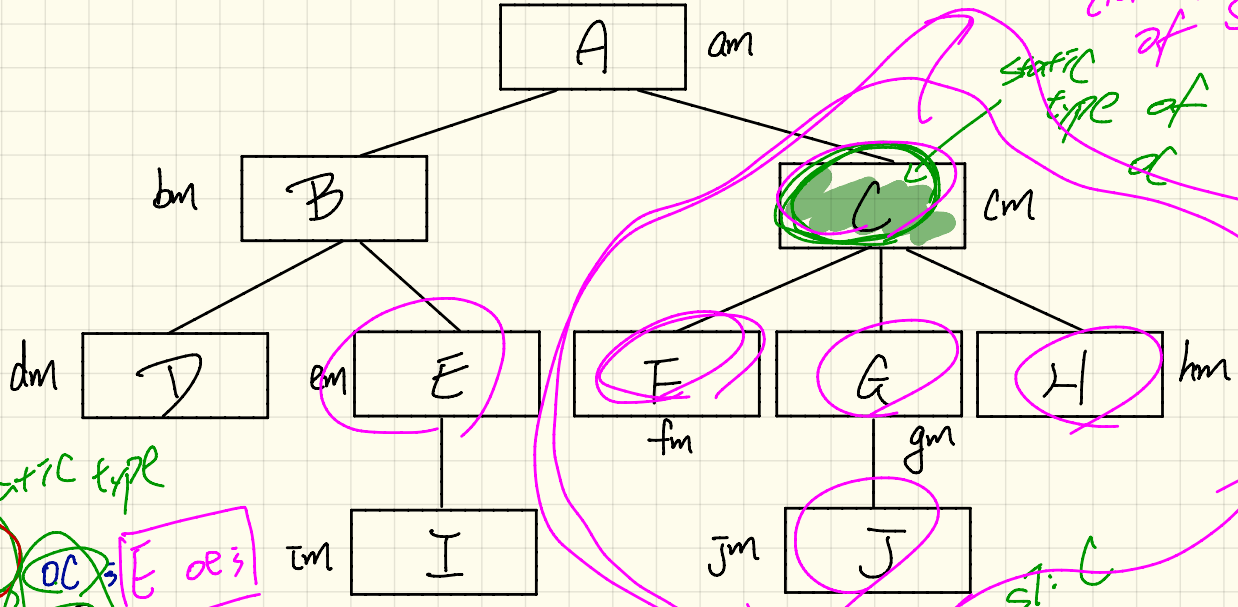


Wednesday Nov. 7
Lecture 17

Static Type vs. Dynamic Type

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

Rules of Substitutions



descendant classes of ST of C

static type of C

static type

- C
- C
- A
- F
- G
- H
- J

OC

OC2

oa

of

og

oh

oj

Safe to substitute OC with:

Unsafe to substitute OC with:

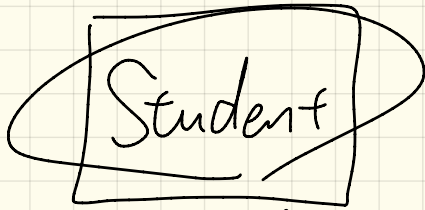
st: C

OC = ?

OC2 of

OC = ?

C OC = OR (E) X

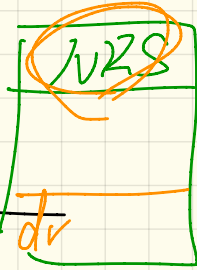
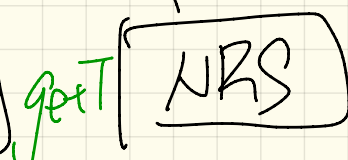
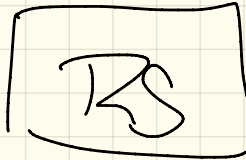


Student S;

RS RS;

NRS NRS;

getT



Polymorphism

S = new RS (); ✓

DT: RS → S.getT();

S = new NRS ();

DT: NRS → S.getT();



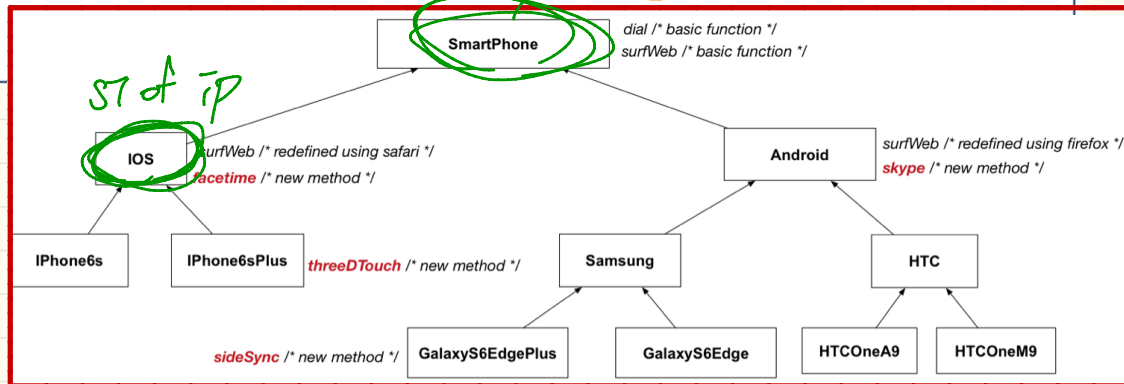
Polymorphism (2)

```
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]

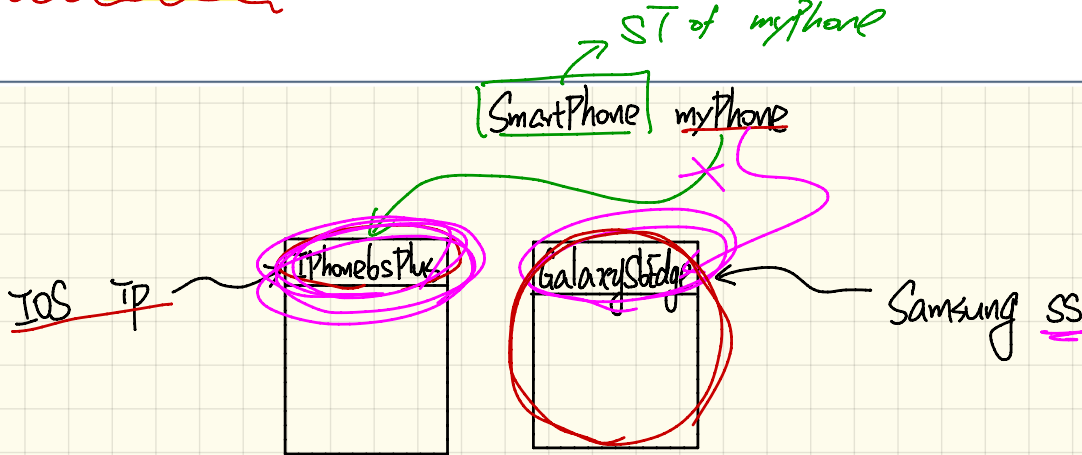
ST of myPhone



Dynamic Binding (2)

```
class SmartPhoneTest2 {  
    public static void main(String[] args) {  
        SmartPhone myPhone;  
        IOS ip = new iPhone6sPlus();  
        myPhone = ip;  
        myPhone.surfWeb();  
        Samsung ss = new GalaxyS6Edge();  
        myPhone = ss;  
        myPhone.surfWeb();  
    }  
}
```

Annotations in the code:
- `SmartPhone` is boxed in green.
- `IOS ip = new iPhone6sPlus();` is circled in red.
- `myPhone = ip;` is circled in red.
- `myPhone.surfWeb();` is boxed in yellow.
- `ST: $P` is written in green above the first `myPhone`.
- `ST: $S` is written in green above the second `myPhone`.
- `DT of ip is IP6sPlus` is written in red above the first `myPhone`.
- `DT of myPhone: iPhone6sPlus` is written in red above the first `myPhone`.
- `DT of myPhone: A6E` is written in red above the second `myPhone`.
- `ST of myPhone` is written in green above the second `myPhone`.



Type Cast: Motivation

```
Student(String name)
void register(Course c)
double getTuition()
```



```
String name
Course[] registeredCourses
int numberOfCourses
```

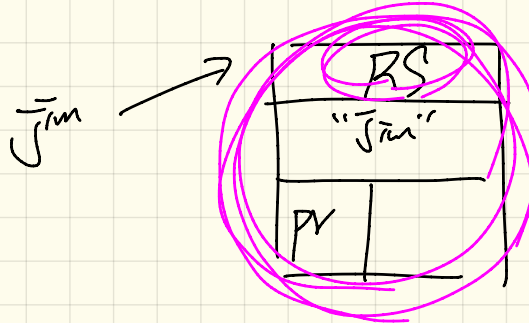
```
/* new attributes, new methods */
ResidentStudent(String name)
double premiumRate
void setPremiumRate(double r)
/* redefined/overridden methods */
double getTuition()
```



```
/* new attributes, new methods */
NonResidentStudent(String name)
double discountRate
void setDiscountRate(double r)
/* redefined/overridden methods */
double getTuition()
```

```
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`.

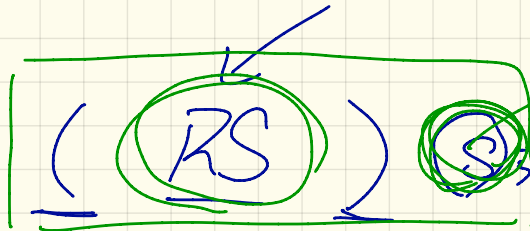
Student S = new RS(...);

RS rs = S; X

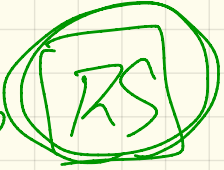


RS

rs =

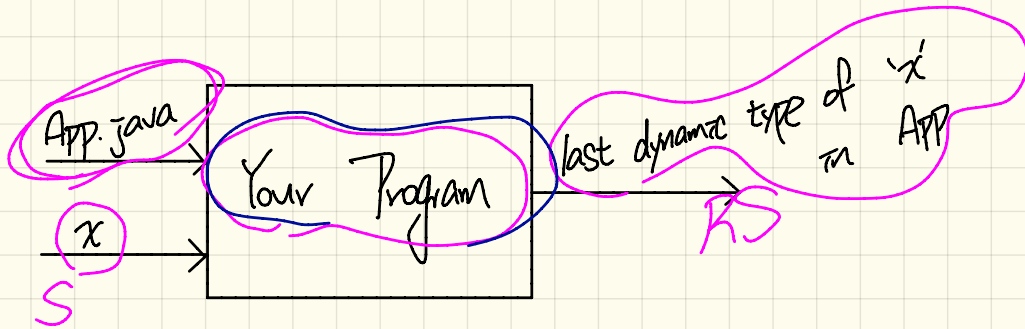


ST:
Student

temporarily change
the ST to 



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

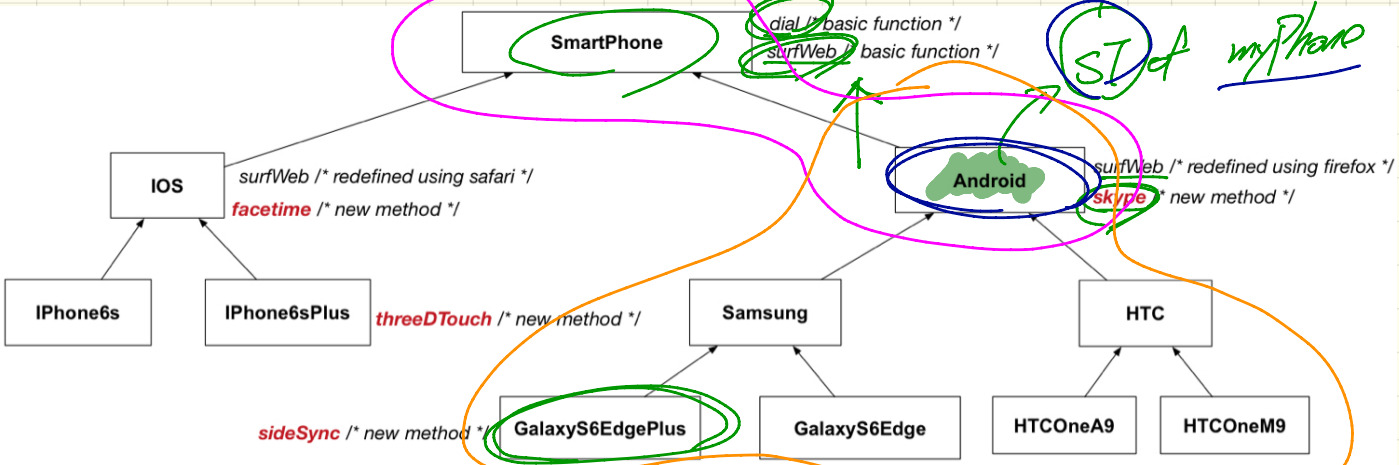
Anonymous Cast

```
SmartPhone aPhone = new iPhone6sPlus();  
((iPhone6sPlus) aPhone).facetime();
```

Problem?

```
1 SmartPhone aPhone = new iPhone6sPlus();  
2 (iPhone6sPlus) aPhone.facetime();
```

Comparable Cast: Upward vs. Downward



EXPECTATIONS

ST of myPhone

```

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

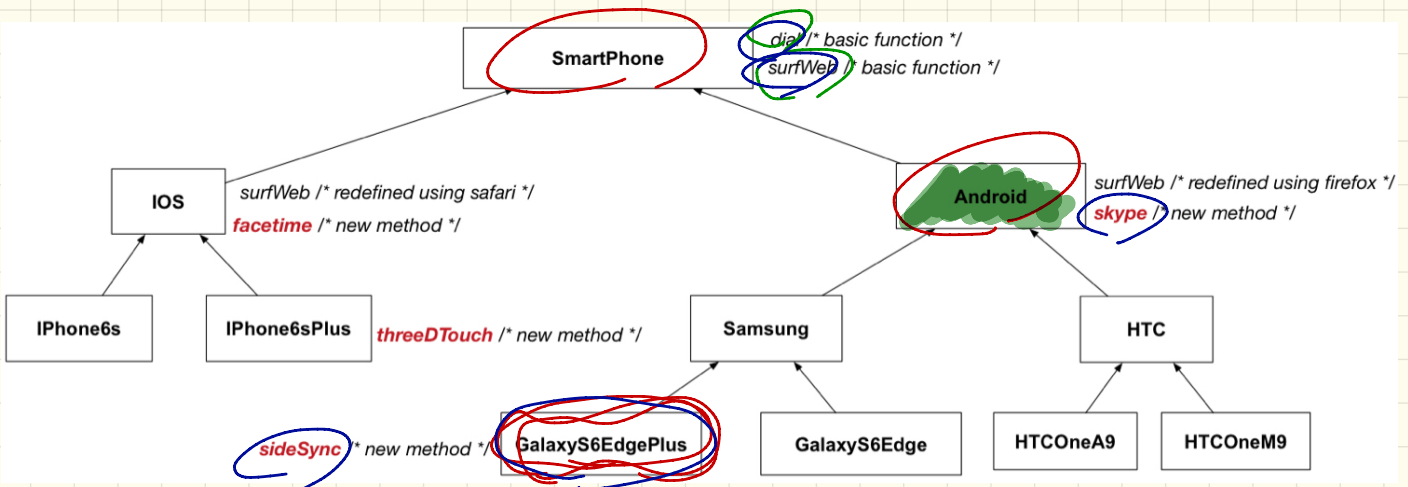
EXPECTATIONS

myPhone : [skype
surfweb
dial

sp : dial
surfweb

ga : dial, surfweb, skype, sideSync

Upward casting
Downward casting

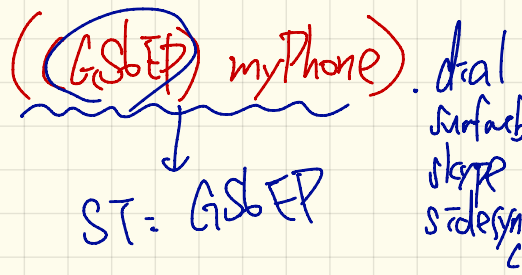
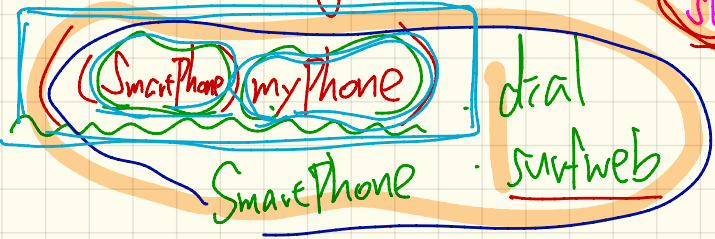


Android

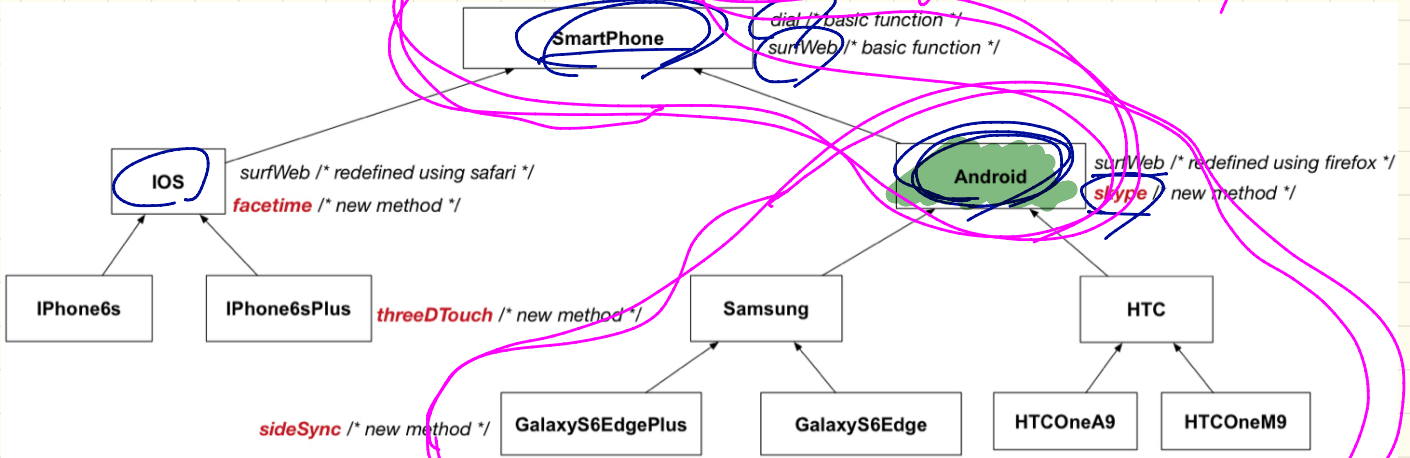
myPhone = ...

Downward Casting

Upward Casting



upward casting (narrow expectation)



Android p =

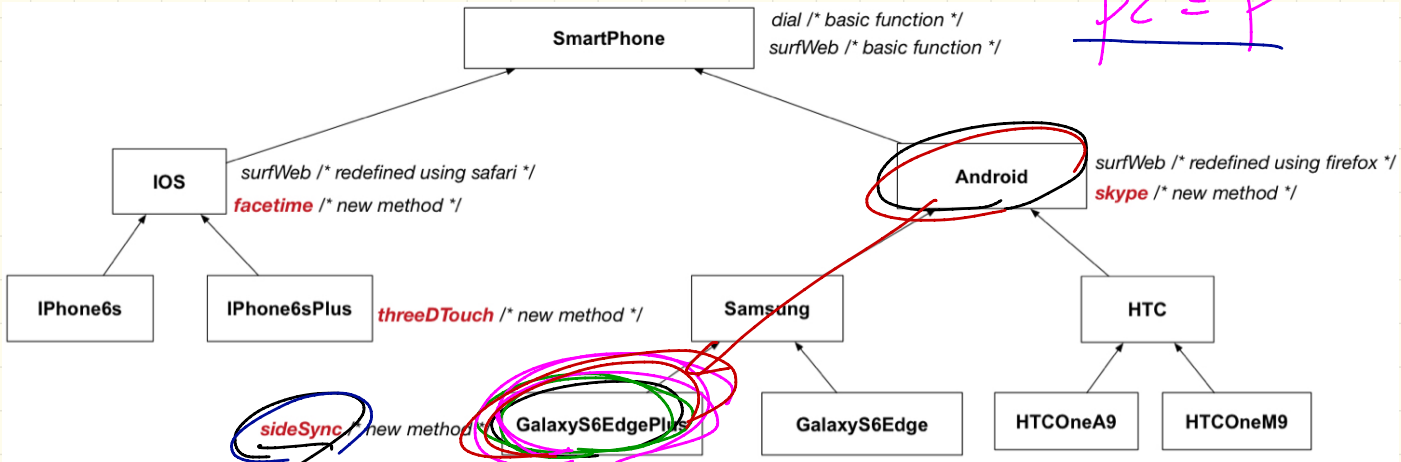
(ST)

p. skype
p. surfweb
p. dial

downward casting
(wider expectation)

What kind of (case) will compile?

↓
temporarily changes the ST.

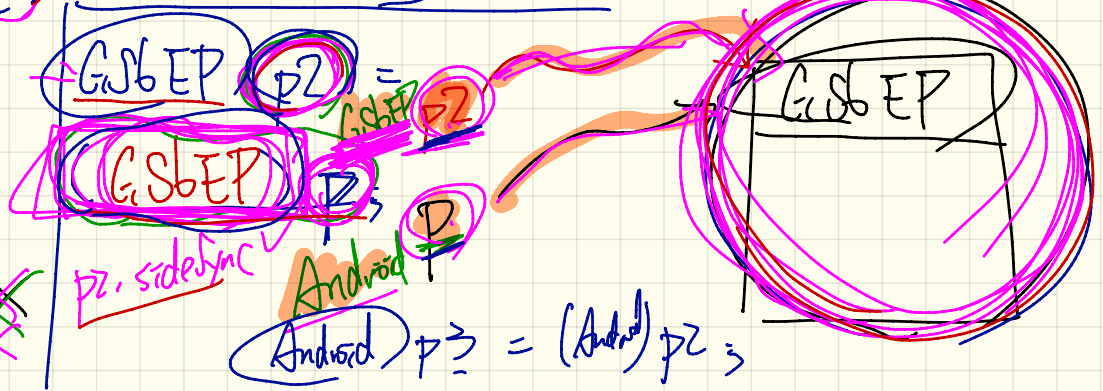


P2 = P

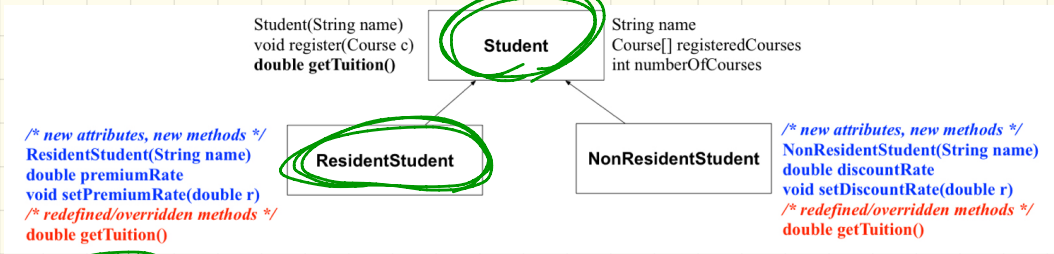
Android P = new GS6EP(...);

- P. skype
- P. surfWeb
- P. dial

~~P. sideSync~~
P. sideSync



Comparable Cast May Fail at Runtime (1)



```

1 Student jim = new NonResidentStudent("J. Davis"); ✓
2 ResidentStudent rs = (ResidentStudent) jim; → downward casting
3 rs.setPremiumRate(1.5); → compile!
  
```

