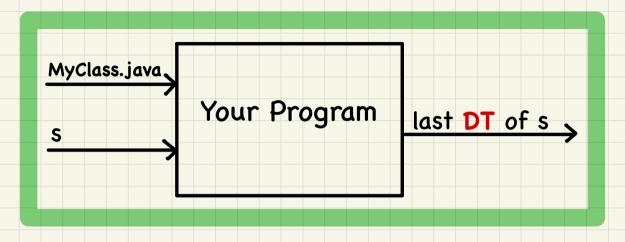
#### Type Cast: Motivation

```
Student(String name)
                                                                            String name
                             void register(Course c)
                                                                            Course[] courses /* registered courses (rcs) */
                                                           Student
                             double getTuition()
                                                                            int noc /* number of courses */
                                                                                                   /* new attributes, new methods */
/* new attributes, new methods */
                                                                                                   NonResidentStudent(String name)
ResidentStudent(String name)
                                                                         NonResidentStudent
                                     ResidentStudent
                                                                                                   double discountRate
double premiumRate
                                                                                                   void setDiscountRate(double r)
void setPremiumRate(double r)
                                                                                                  /* redefined/overridden methods */
/* redefined/overridden methods */
                                                                                                   double getTuition()
double getTuition()
```

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

# An A+ Challenge: Inferring the DT of a Variable



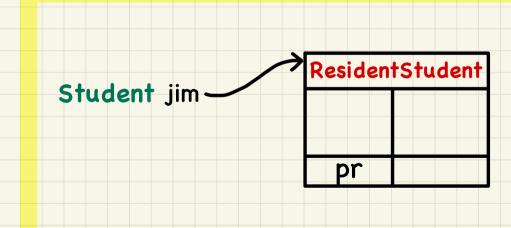
```
class MyClass {
    main (...)
    Student s = ...;
    ...
    s = new ResidentStudent(...);
}
```

# Anatomy of a Type Cast

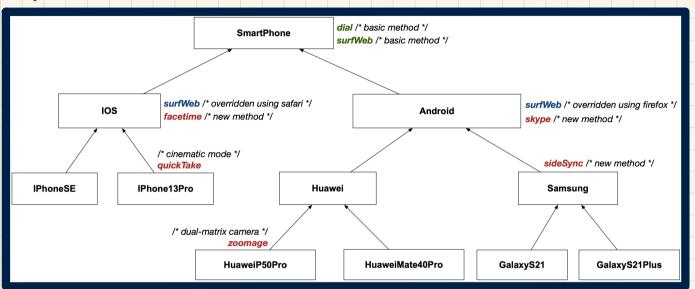
Student jim = new ResidentStudent("Jim");

```
ResidentStudent rs = (ResidentStudent) jim;

an alias whose ST is ResidentStudent
```



# 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

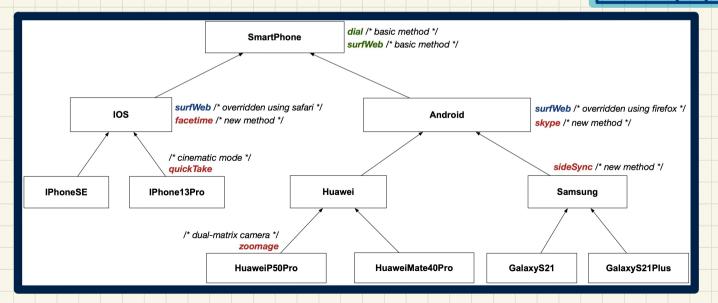
Android myPhone = new GalaxyS21Plus();

**SmartPhone** sp = (**SmartPhone**) myPhone;

**GalaxyS21Plus** ga = (**GalaxyS21Plus**) myPhone;

Expectations

	cn	myPhone	04
	٩٢	IllyPhone	gu
dial			
surfWeb			
skype			
sideSync			
facetime			
quickTake			
zoomage			

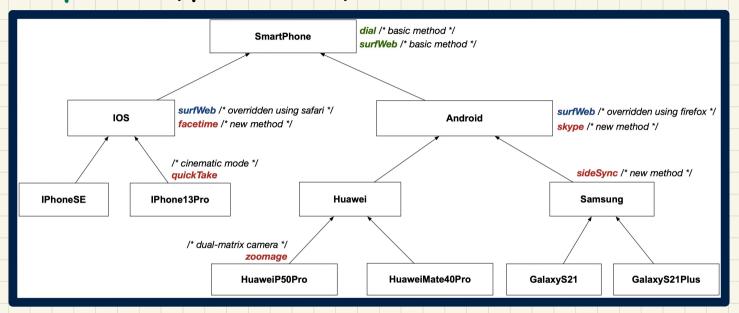


### Compilable Type Cast May Fail at Runtime (1)

```
Student(String name)
                                                                             String name
                                                                             Course[] courses /* registered courses (rcs) */
                              void register(Course c)
                                                            Student
                              double getTuition()
                                                                             int noc /* number of courses */
                                                                                                   /* new attributes, new methods */
/* new attributes, new methods */
                                                                                                   NonResidentStudent(String name)
ResidentStudent(String name)
                                                                          NonResidentStudent
                                      ResidentStudent
                                                                                                   double discountRate
double premiumRate
                                                                                                   void setDiscountRate(double r)
void setPremiumRate(double r)
                                                                                                   /* redefined/overridden methods */
/* redefined/overridden methods */
                                                                                                   double getTuition()
double getTuition()
```

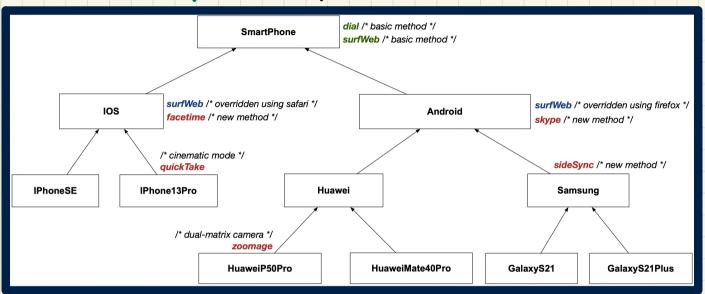
```
1 Student jim = new NonResidentStudent("J. Davis");
2 ResidentStudent rs = (ResidentStudent) jim;
3 rs.setPremiumRate(1.5);
```

#### Compilable Type Cast May Fail at Runtime (2)



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

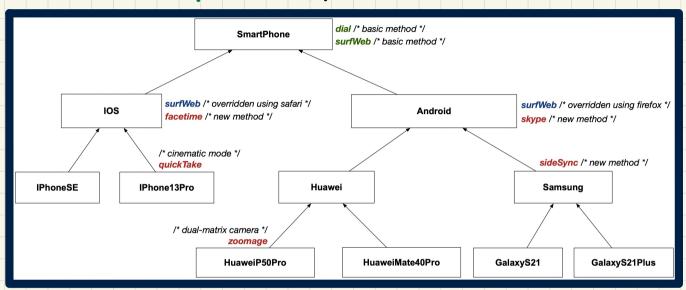
# Exercise: Compilable Type Cast? Fail at Runtime? (1)



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

# Compilable? ClassCastException at runtime?

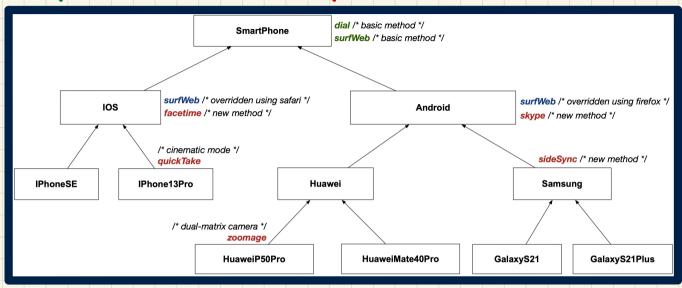
# Exercise: Compilable Type Cast? Fail at Runtime? (2)



```
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

Non-Compilable Casts

Exception-Free 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();
D d = (D) b:
```

