

Static vs dynamic types

- Static type (know it without "running")
 - The type of the variable
 - This is what the compiler knows
 - Dynamic type
 - The "full" type of the actual object
 - This is known only at runtime
- Object o = new String("hello");

	Legal?	Static Type	Dynamic Type
arrL.add("Hi");		xxx	xxx
arrL.add(new Dog());		xxx	xxx
arrL.add(new Cat());		xxx	xxx
arrL.add(42);		xxx	xxx
arr[0] = "Hi";		xxx	xxx
arr[1] = new Dog();		xxx	xxx
arr[2] = new Cat();		xxx	xxx
arr.get(0)	xxx		
arr.get(1)	xxx		
arr[2]	xxx		
c = arrL.get(2);		xxx	xxx
c = arr[2]		xxx	xxx
d = (Dog)arrL.get(1)		xxx	xxx
c = (Cat)arrL.get(1)		xxx	xxx
arr[1].eat()		xxx	xxx
arr[2].eat()		xxx	xxx
arr[1].bark()		xxx	xxx
d.bark()		xxx	xxx

Overriding

```

class Animal {
    void eat() {
        System.out.println("Animal eat");
    }
};
class Dog extends Animal {
    void eat() {
        System.out.println("Dog eat");
    }
}
class Cat extends Animal {
    void eat() {
        System.out.println("Cat eat");
    }
}
Animal a,a1,a2;
Dog d;
Cat c;

a = new Animal();
d = new Dog();
a1 = d;
c = new Cat();
a2 = c;
    
```

Overloading

```

Class Dog {
    public void eat();
    public void eat(String food);
    public void eat(Cat ct);
}
    
```

Overloading

```

Class Dog {
    public void eat();
    public boolean eat(String food);
    public void eat(Cat ct);
}
    
```

Overloading: tricky

```

Class Employee {
    public double getPay();
    public boolean getPay(); Illegal!!
}
    
```

Overloading(trickier)

```
Class Dog {  
    public void eat();  
    public void eat(String food);  
    public void eat(Cat ct);  
    public void eat(Animal an);  
}  
Cat c    = new Cat();  
Animal a = c;  
eat(c)  
eat(a)
```

Unit Test

```
public class HourlyEmployee {  
    int hours;  
    int payRate;  
    ...  
    public static void test() {  
        new HourlyEmployee("name",40,10);  
        int totalPay = getPay();  
        if (totalPay != (40*10)) {  
            System.out.println("Wrong pay");  
        }  
    }  
}
```

Java Doc

```
/**  
 * This class represents a dog.  
 *  
 * @author Eric Lemar  
 */  
public class Dog {  
    /**  
     * Tell the dog to bark.  
     *  
     * @param times The desired number of barks  
     * @return Return false if refused to bark, true otherwise  
     */  
    public boolean bark(int times) {}  
}
```