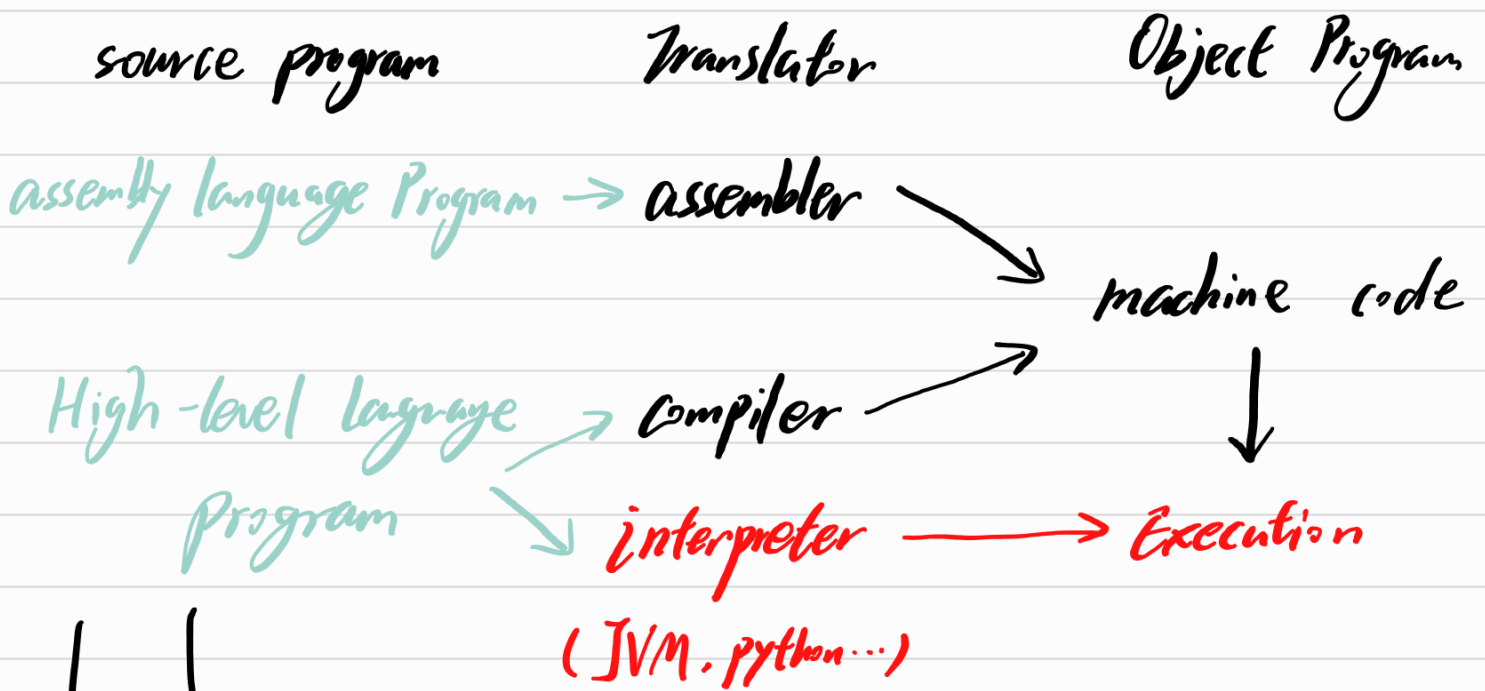


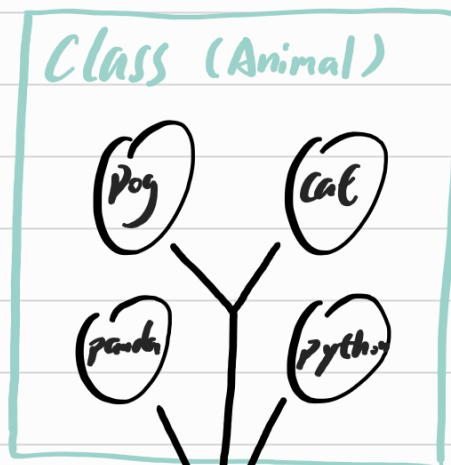
Translators



procedural : a series of steps and procedures
languages

object-oriented : made of objects
languages

object is an instance of
class.



class is a blueprint
from which objects are made

data : instance fields
(state)

objects

behaviors: methods

object-oriented: ① PRY: Do not repeat yourself.

② Shy: should not reveal the info about itself unless necessary

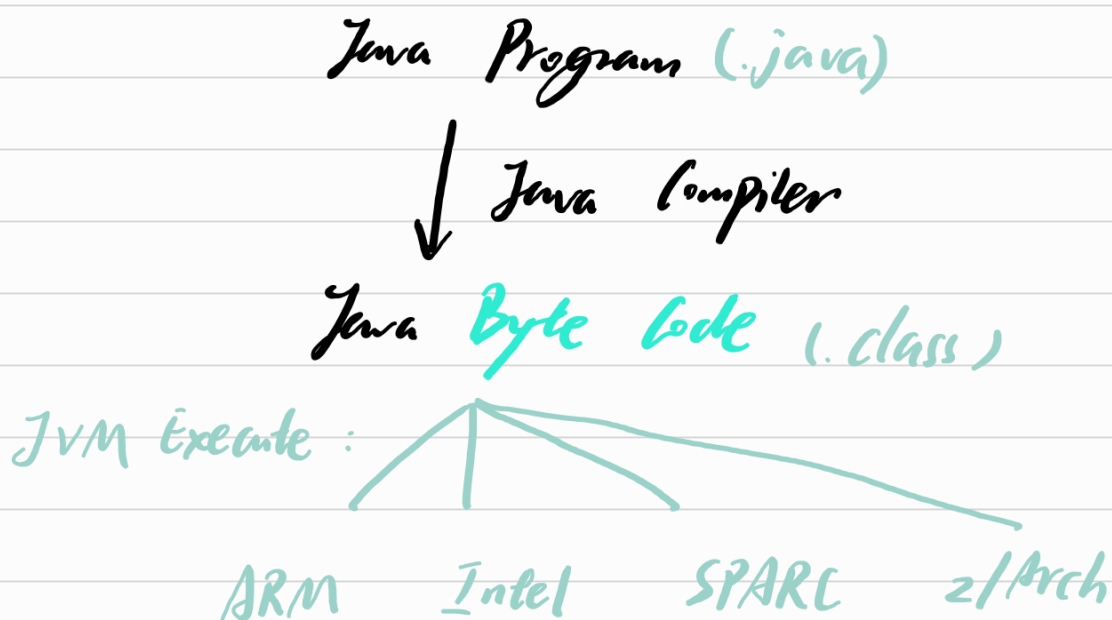
③ Send a message than doin' func call.

Benefits: ① Modularity: object source code can be maintained independently for other objects

② Information-hiding

③ Code-reuse

Java Compilation Model



Heart of Java: ① Java Virtual Machine (JVM)
② Garbage Collection

Naming Convention: Variables & Method name:
begin w/ lower case letters

Class name:
begin w/ higher case letters

Java do not have global variables

Method overloading (Ad-hoc polymorphism)

Same name, diff parameter list / return type

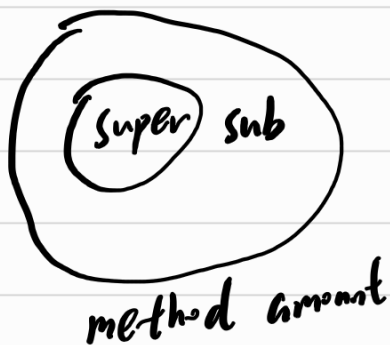
@ Override : ask Java to check for errors

Access Modifiers	same class	same package	subclass	other
- private	✓	✗	✗	✗
no modifier	✓	✓	✗	✗
# protected	✓	✓	✓	✗
+ public	✓	✓	✓	✓

(only public & no modifier can be used in class declaration)

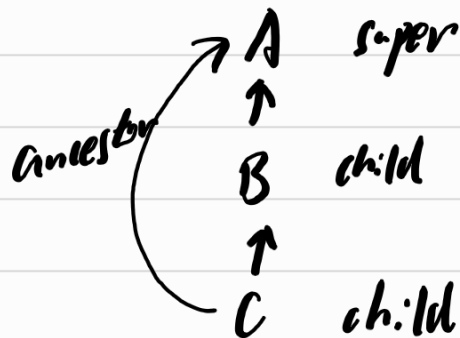
Encapsulation

Inheritance : super class



sub classes extends super class

(multi inheritance is not allowed in Java)



⇒ Constructor inheritance:

For child ⇒ `super( )`,

For same class ⇒ `this( )`.

"final" Modifier: `final int x = 3`

x cannot be changed / redefined

⇒ for a final class, it can't have sub-class.

"Object" class: Every class in Java is a descendant of object class

⇒ a (object) parameter can be replaced
by an object of any class
(subtype polymorphism)

```
⇒ toString() {  
    return ..... ;  
}
```

⇒ object 1. equals (object 2)

Dynamic Binding : search from this class to object class

Overloading : same class & same name & diff signature

Override : sub class & same name & same signature