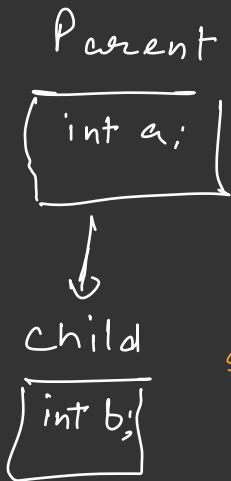


# Pillars of OOPS

- ↳ ① Inheritance
- ② Polymorphism
- ③ Encapsulation
- ④ Abstraction

## ① Inheritance

↳ Same as inheriting properties from parent to child



child class can inherit the properties of parent

Syntax

```
class Parent {  
    int a;  
    void printA() {  
        syso(a);  
    }  
}
```

```
class Child extends Parent {  
    int b;  
    void printB() {  
        syso(b);  
    }  
}
```

Child c = new Child c)

c.b

c.print BC)

c.a

c.print AC)

## Types of Inheritance

① Single Level ⇒



② Multilevel ⇒

class A {

int a;

}

class B extends A {

int b;

}

class C extends B {

int c;

}



main()

C obj = new C();

obj.a,

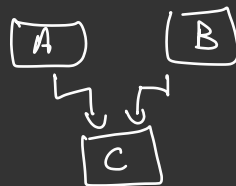
obj.b;

obj.c;

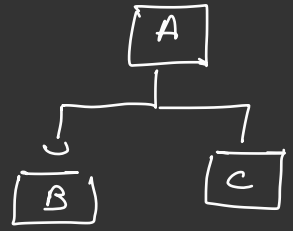


③ Multiple Inheritance

Not allowed in java  
classes

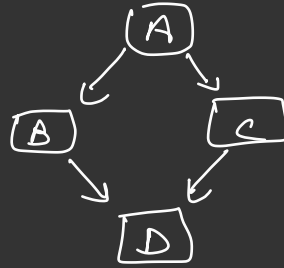


## ④ Hierarchy Inheritance



## ⑤ Hybrid Inheritance

not allowed

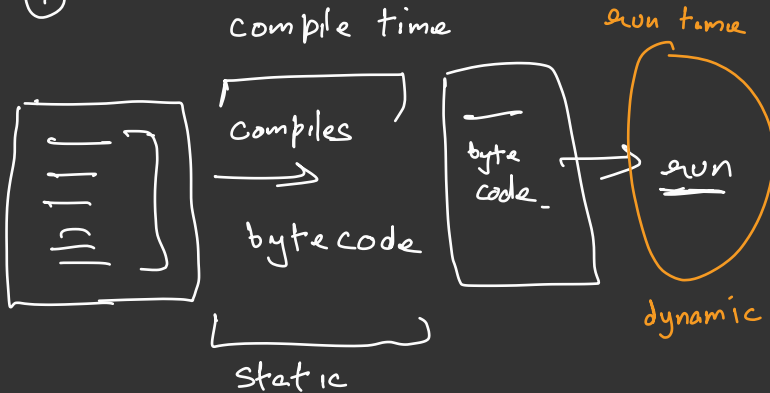


## ② Polymorphism

many

form

①



① Compile time / Static

② Run time / Dynamic

→ Compile time / static

function overloading

```
void sum (int a, int b) {  
    syso (a+b)  
}
```

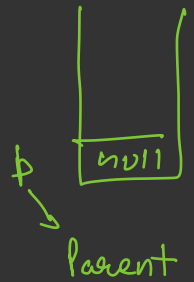
```
void sum (int a, int b, int c) {  
    syso (a+b+c);  
}
```

```
void sum (double a, double b) {  
    syso (a+b);  
}
```

```
main() {  
    sum(2, 3) //  
    sum(2, 3, 4)  
    sum(2.0, 3.0)  
}
```

Parent (p) = now Parent()

Parent a = new Child();



Run time Polymorphism / Dynamic

---

Final key word

final void area() {  
}

→ Prevents Overriding

final class Shape {

}  
→ Prevent inheritance