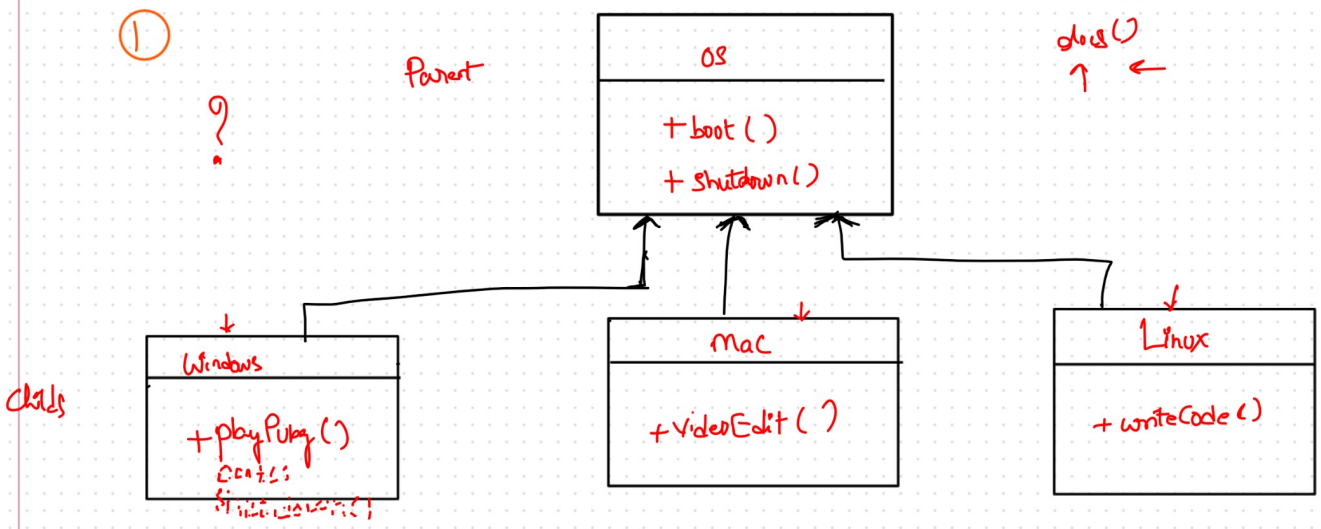
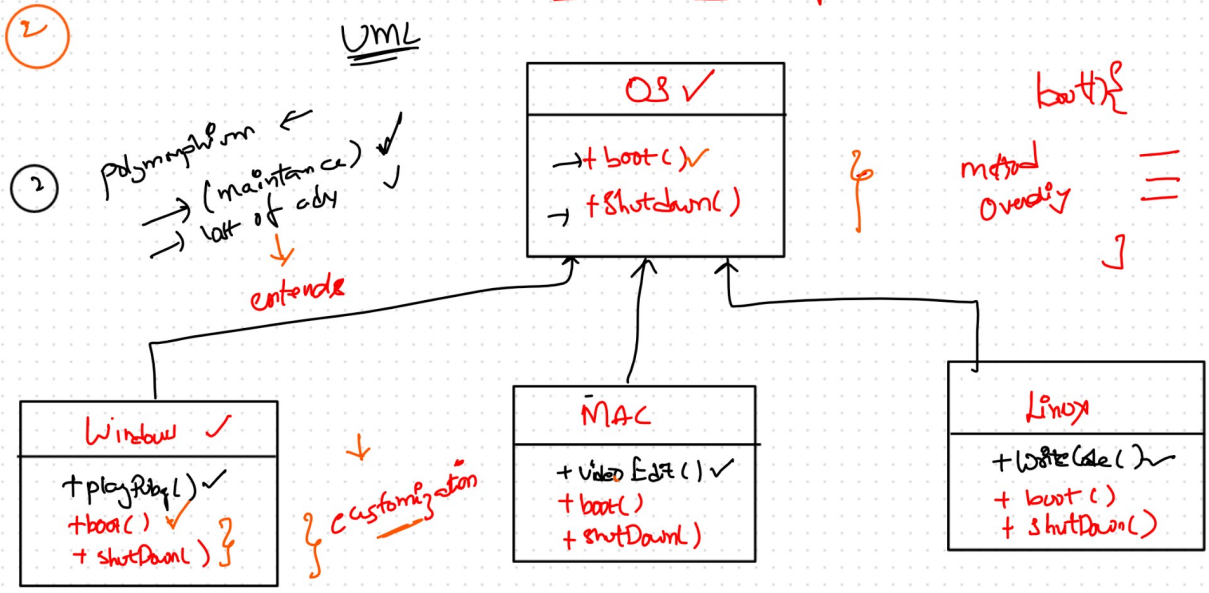


Method Overriding



Note:

1) The methods which are defined within that particular class becomes "specialized methods" for that particular class respectively

eg: +playPubg(), +videoEdit(), +writeCode()

2) if a method is inherited from the parent class & if the body of the method is changed in the child class then such methods we call it as overriden method (or method overriden)

eg: +boot(), shutDown() // overriden method

~ ↓↓ if not customized

eg: Windows, Mac → inherited method (look for prototype-1)

Object class

→ Object class is the parent class for all the classes present/written by users in java.

Prototype, blueprint, type, class } same

→ If we create a class & if we don't use extends (keyword) then that class would inherit Object by default

```
class A {
    p.s.v.m(){
    }
}
```

class A extends Object {

Inheritance

methods of Object class:

class → className, address, hashing, clone

java.

(1) getClass()

public final native java.lang.Class<?> getClass(); (fully qualified method in package)

→ this method returns the fully qualified class Name

```
class ExplainingObjectClass{

}

class Main {
    public static void main(String[] args) {
        ExplainingObjectClass obj = new ExplainingObjectClass();
        System.out.printf("Class Name is %s",obj.getClass());
    }
}
```

implicitly extends Object

// class ExplainingObjectClass

Multi-level Inheritance

A → Object methods, add

B → sub()

C { add, sub }

(2) toString(): this method always implicitly called by System.out.println/printf. this method returns fully qualified className & address of object & we can override to our needs

eg:

```
class ExplainingObjectClass extends A{
    public String toString(){
        return "This class belongs to ExplainingObjectClass"
    }
}

class Main {
    public static void main(String[] args) {
        ExplainingObjectClass obj = new ExplainingObjectClass();
        System.out.println(obj); //ExplainingObjectClass@15db9742
                                 //This class belongs to
                                   ExplainingObjectClass
    }
}
```

String

public java.lang.String toString()

↑ pkg.name

import java.lang.*;

toString()
Integer