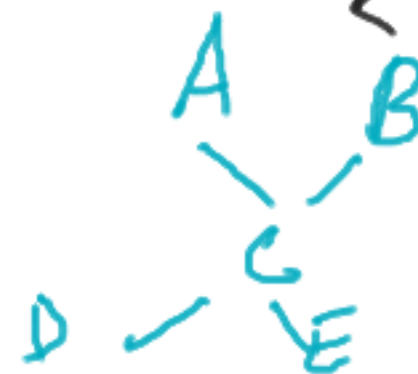
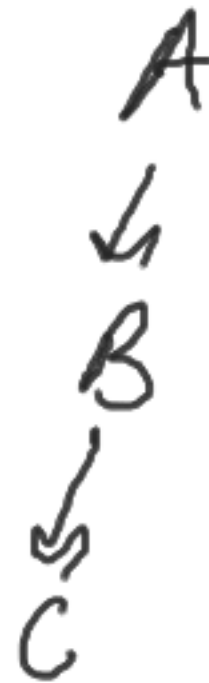


Interface :- almost similar to abstract class.

→ multiple Inheritance



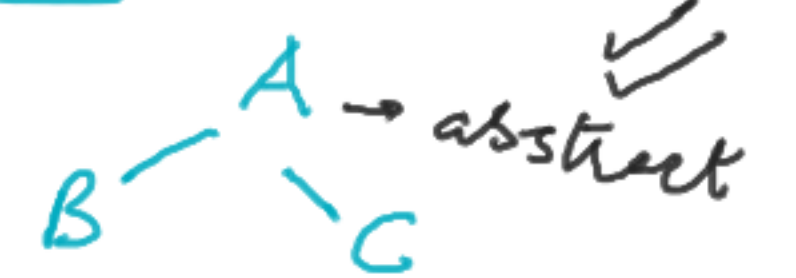
→ single level ✓

→ Multi Level

→ Multiple - A B  
C

→ Hybrid

→ Hierarchical

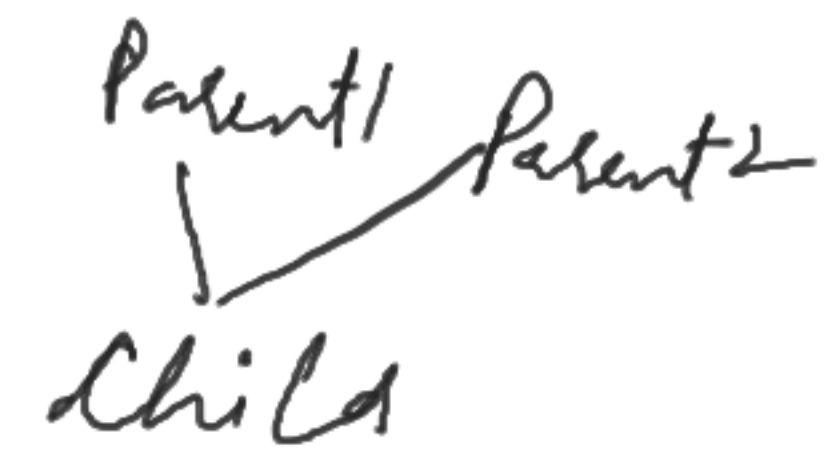


Non-abstracts

```
Class A {  
  int a; 10  
}
```

```
Class B {  
  int a; 20  
  int b;  
}
```

class C extends A, B {  
 }  
X



Multiple

Inheritance

can't achieve

without interfaces.

a  
10  
a  
20

}

# Interface :-

15K	1.7
	1.87

→ variables / fields /  
properties  
by default

"static final"

Can't create object  
interface I, {  
↓  
keyword

→ all the methods  
should be  
abstracted  
- There is no  
need to use  
"abstract" keyword

Can't create  
object  
abstract class A {

final int <sup>N.S variable</sup> a;

→ one time in constructor  
we can assign the value

static final double (PI) = 3.14;  
int b = 10.0;

↓  
CONSTANT

↓  
initialization is mandatory

Class extends class

class implements interface

interface extends interface





Browser

The diagram illustrates the Selenium WebDriver architecture. On the left, a vertical line represents the WebDriver interface, with arrows pointing to specific browser implementations: Chrome, FF (Firefox), Edge, and IE. On the right, a circle represents the WebDriver interface, with the word 'interfaces' written next to it. Inside the circle, the method 'get()' is written and underlined. An arrow points from the 'get()' method to the Selenium WebDriver interface. Another arrow points from the Selenium WebDriver interface to the 'get()' method. A third arrow points from the Selenium WebDriver interface to the 'get()' method. A fourth arrow points from the Selenium WebDriver interface to the 'get()' method.