* **Multi-level:**

```
class A {
    methodA();
}
class B {
    methodB();
}
class C {
    methodC();
}
```

→ obj → A   mA() ✓

→ obj → B   mA() ✓ mB() ✓

obj → C   mA() ✓ mA() ✓ mC() ✓

Abstraction: →

Showing the functionality (necessary data) & hiding the implementation details from the end user is called data abstraction.

Showing → What is happening (functionality)

hiding → how it is happening (implementation)

(I) Abstract Classes & Methods (0 – 100%)

(II) Interface → Java → C++ (Pure Virtual Function)
100%

Better UX

* Polymorphism: The process by which same entity can behave diffently.

Entity:		Place	Role
	Mohit	Classroom	Student
		Restaurant	Customer
		Home	Son

Poly + Morph
↓ ↓
Many forms/shapes

(I)	static	compile time	overloading	Same class
(II)	dynamic	run time	overiding	Multiple Classes