Java v.s. C

- All functions in Java are inside one or the other Class

- While calling any function, we have to use dot (.) and Class name / variable before it to specify functions of which class we want to call.

- Non local variables are also accessed with a dot (.)

Public and Private

- If something is private, it can only be accessed in same Class in which it is defined.
- Used for all functions and all non local variables.
- We try to make private as many things as possible.
- This is called Abstraction.

Static and Non Static

- Static - Defined for class
- Non Static - defined for individual objects/ instances of Class

Inheritance

- Every Class (except Object class) extends some class.
- This means that all methods*/functions and variables/attributes of parent class are inherited by Child class. So a child class is **parent class + more**
- * constructor functions are inherited in one special case only.

Constructor

- A **non static** function in a class with **same name** as Class and **no return type**.
- Every class must have at least one constructor.*
- Used to set initial state of newly created objects.

Method Overloading

- Multiple functions in same Class with same name but different parameter type.
- If one function is inherited and one implemented, that also counts as Overloading

Method Overriding

- Creating a function with same name and **same parameter type** in subclass
- If you create a function with same name but different parameter in subclass that's not overriding. That's overloading.

what is **this**?

- Found only inside non static functions.
- In constructors, it represents newly created object.
- In other functions, it represents the object on which method was called.

Interface

- List of functions, without implementation.
- Used to ensure certain behaviour in Classes.

Variable v.s. Object

- Object is the real thing, while variable is just a way to use the object.

- A Variable of XYZ class can store objects of XYZ class and it's subclasses.

- A variable of ABCD interfaces can store objects of classes* implementing ABCD interface or it's sub interfaces.

- *class can inherit the implementation of interface.