Dependency Injection

Video source - <https://www.youtube.com/watch?v=yunF2PgJlHU>

It is a way of designing code (aka creating objects), in such a way that, the creation of dependencies that a main object depends upon is outsourced to a third app or the creator app!

Giving below example to explain above –

Lets say we have a Computer Object which is dependent on RAM, CPU, Motherboard, Storage.

We can declare computer class as below –

Class Computer {

class RAM,

class CPU,

class Motherboard,

class Storage

}

Now if we declare the above design, we would need to initialize the Computer Object as below –

Computer HPComp = new Computer()

Then to initialize the values we would need to do something like this –

HPComp.RAM.size = 10 GB

HPComp.RAM.company = Intel .. etc

This would mean that the class Computer is dependent on class RAM as well as, class computer also needs to know internal implementations of class RAM.

This would mean that any internal changes in class RAM, the class Computer needs to be aware of.

So the person, application creating Computer Class must also need to know the internal implementation of the class RAM.

This requirement of needing to know the internal implementations of class RAM can be taken care of if we design the code in below way –

Class Computer {

Computer(RAM RAMObject)

{

This.RAM = RAMObject

}

}

Here, what we have done is we have declared a contructor which would taken in RAM object and initialize the Computers RAM object.

In this style of design, Computer Class need not know the internal implementation of its dependent class RAM, and that creation of RAM object can be done separately as composed to Computer object.

This is called as dependency injection.

Maybe more on this can be researched later!!