**Day 13:**

**Design pattern :** Design pattern is best practise or solution for repeating problem.

Core Design pattern : GOF

23 design pattern.

These design pattern mainly divided into 3 types.

1. **Creational design pattern**: these design pattern deal with object creation.
2. Singleton design pattern:
3. Builder design pattern
4. **Factory pattern**
5. **Abstract factory pattern**
6. Prototype design pattern
7. **Structural design pattern:** structure of class and interfaces.
8. Adapter design pattern
9. Composite
10. Flyweight design pattern
11. Proxy design pattern
12. Decorator design pattern
13. Bridge design pattern
14. Façade design pattern

1. **Behavioural design pattern:** these design patter deal with object interaction.
2. Chain of responsibility:
3. Command design pattern
4. Visitor design pattern
5. Template design pattern
6. Interpreter design pattern
7. Mediator design pattern
8. Memento design pattern
9. Iterator design pattern
10. Observable design pattern
11. State design pattern
12. Strategy design pattern

Composite design pattern :Composite design pattern describe as compose object into tree structure to represent part whole hierarchies.

Book has an author

Library has many books

class Book {

Book class property

Author a = new Author();

}

class Author {

Author property

}

class Library {

Library class properties

List<Book> listOfBooks =new ArrayList();

}

**Proxy design pattern :** proxy design pattern provide subsite or place holder for another object to control the access and functionality

MOM : message oriented middleware tools or technologies or product etc

P2P : Point to Point 🡪 One to One communication

Pub/Sub : Publisher and Subscriber : one to many relationship

**Observable design pattern**

: It is a type of behavioural design pattern. It support features as one to many relationship.

**Strategy design pattern :** It is a type of behaviour design pattern. Which contains more than one strategy and on demand we can change the strategy.

**Command design pattern:** Encapsulate a request as object for parameterization for queue, stack, logging where we can do the operation as undo.