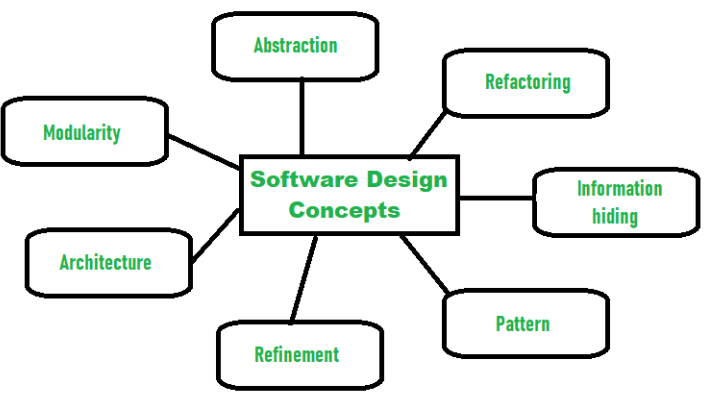
**Software design: software design concepts**

Software Design Concepts:  means the ideas or principles behind software design. It describes how you plan to solve the problem of designing software, the logic, or thinking behind how you will design software. It allows the software engineer to create the model of the system or software or product that is to be developed or built. The software design concept provides a supporting and essential structure or model for developing the right software. There are many concepts of software design and some of them are given below:



**Abstraction- (Hide Irrelevant data )**Abstraction simply means to hide the details to reduce complexity and increases efficiency or quality. Different levels of Abstraction are necessary and must be applied at each stage of the design process so that any error that is present can be removed to increase the efficiency of the software solution and to refine the software solution. The solution should be described in broad ways that cover a wide range of different things at a higher level of abstraction and a more detailed description of a solution of software should be given at the lower level of abstraction.

**Modularity- (subdivide the system)**Modularity simply means dividing the system or project into smaller parts to reduce the complexity of the system or project. In the same way, modularity in design means subdividing a system into smaller parts so that these parts can be created independently and then use these parts in different systems to perform different functions. It is necessary to divide the software into components known as modules because nowadays, there are different software available like Monolithic software that is hard to grasp for software engineers. So, modularity in design has now become a trend and is also important. If the system contains fewer components then it would mean the system is complex which requires a lot of effort (cost) but if we are able to divide the system into components then the cost would be small.

**Architecture- (design a structure of something )**Architecture simply means a technique to design a structure of something. Architecture in designing software is a concept that focuses on various elements and the data of the structure. These components interact with each other and use the data of the structure in architecture.

**Refinement- (removes impurities)**Refinement refers to the process of removing any impurities if present and hence increasing the quality of software. The refinement concept of software design is actually a process of developing or presenting the software or system in a detailed manner that means to elaborate a system or software. Refinement is very necessary to find out any error if present and then to reduce it.

**Pattern- (a Repeated form)**The pattern simply means a repeated form or design in which the same shape is repeated several times to form a pattern. The pattern in the design process means the repetition of a solution to a common recurring problem within a certain context.

**Information Hiding – Hide the Information**Information hiding simply means to hide the information so that it cannot be accessed by an unwanted party. In software design, information hiding is achieved by designing the modules in a manner that the information gathered or contained in one module is hidden and can’t be accessed by any other modules.

**Refactoring-( Reconstruct something )**Refactoring simply means reconstructing something in such a way that it does not affect the behavior of any other features. Refactoring in software design means reconstructing the design to reduce complexity and simplify it without impacting the behavior or its functions. Fowler has defined refactoring as “the process of changing a software system in a way that it won’t impact the behavior of the design and improves the internal structure”.

**Others include:**

**Coupling and cohesion:** Coupling measures the degree of dependency between different modules or components in a software system. Loose coupling is desirable as it promotes flexibility and maintainability. while Cohesion measures how closely the elements within a module or component are related. High cohesion within a module means that its elements work together to achieve a common purpose.

**Concurrency:** This in software refers to the execution of multiple tasks or processes simultaneously. It is particularly relevant in systems where multiple threads, processes, or tasks run concurrently, sharing resources and potentially executing in overlapping time intervals. Concurrency introduces complexities, and managing it effectively is crucial for creating efficient and responsive software. Here are key considerations related to concurrency

**Design verification:** This involves the process of ensuring that a software design meets its specified requirements and adheres to the intended design principles. It is a critical step in the software development life cycle to catch errors, validate functionality, and confirm that the design aligns with the desired system behavior. Key aspects of design verification include: requirements verification, formal methods, code reviews, static analysis etc