

# Chapter One

## Introduction to software design

# What is software design?

- ✓ Software Design is the process of transforming user requirements into a suitable form, which helps the programmer in software coding and implementation.
- ✓ During the software design phase, the design document is produced, based on the customer requirements as documented in the SRS document.

# Cont...

- **The following items are designed and documented during the design phase:**
  - Different modules are required.
  - Control relationships among modules.
  - Interface among different modules.
  - Data structure among the different modules.
  - Algorithms are required to be implemented among the individual modules.

# Objective of software design

- **Correctness:** A good design should be correct i.e., it should correctly implement all the functionalities of the system.
- **Efficiency:** A good software design should address the resources, time, and cost optimization issues.
- **Flexibility:** A good software design should have the ability to adapt and accommodate changes easily. It includes designing the software in a way, that allows for modifications, enhancements, and scalability without requiring significant rework or causing major disruptions to the existing functionality.
- **Understandability:** A good design should be easily understandable, it should be modular, and all the modules are arranged in layers.
- **Completeness:** The design should have all the components like data structures, modules, external interfaces, etc.
- **Maintainability:** A good software design aims to create a system that is easy to understand, modify, and maintain over time.

# Software Design Activities

- ✓ Software Design is the process of creating the blueprint for the Software system being build. Generally, there are 3 major activities in Software Design –
  - External Design
  - Architectural Design
  - Detailed Design
- ✓ The Architectural design and Detailed design are collectively known as Internal Design.
- ✓ **External Design** – External design involves conceiving, planning out and specifying the externally observable characteristics of Software Product such as User interface, report formats, data sources, functional characteristics and performance requirements.

# Cont...

- ✓ **Internal Design** – Internal design involves conceiving, planning out and specifying the internal structure and processing details of the software product. The goals of internal design are to specify internal structure and processing details, to record design, to elaborate the test plan and to provide alternatives, provide the blueprint design for implementation, specifying using and maintenance activities.
  - ✓ **Architectural Design** – Architectural Design involves the refining of the conceptual view of the system, identifying data entities, attributes, internal processing flows and their relationships.
  - ✓ **Detailed Design** – Detailed design include specifications of algorithms/activity flows and detailed class diagrams that provide detailed blueprints for concrete implementation of data models/objects and their interrelationships.

# Design considerations

- ✓ When designing a piece of software, there are many aspects to be considered. What is expected from the software will define the importance of each point of consideration.
  - **Usability:** The end-user must be able to easily interact with the UI. Location of UI elements, colors and default value for parameters are factors to be considered.
  - **Extensibility:** When adding new capabilities to the software, it should be done without significant changes to the architecture.
  - **Compatibility:** How is the software going to interact with other products. For example backward-compatibility with an older version of itself
  - **Modularity:** Various components of the software could be implemented and tested independently.
  - **Maintainability:** How easy is to apply bug fixes and modifications to the software
  - **Functionality:** How well the software can execute the user tasks
  - And many more...

# Design principles

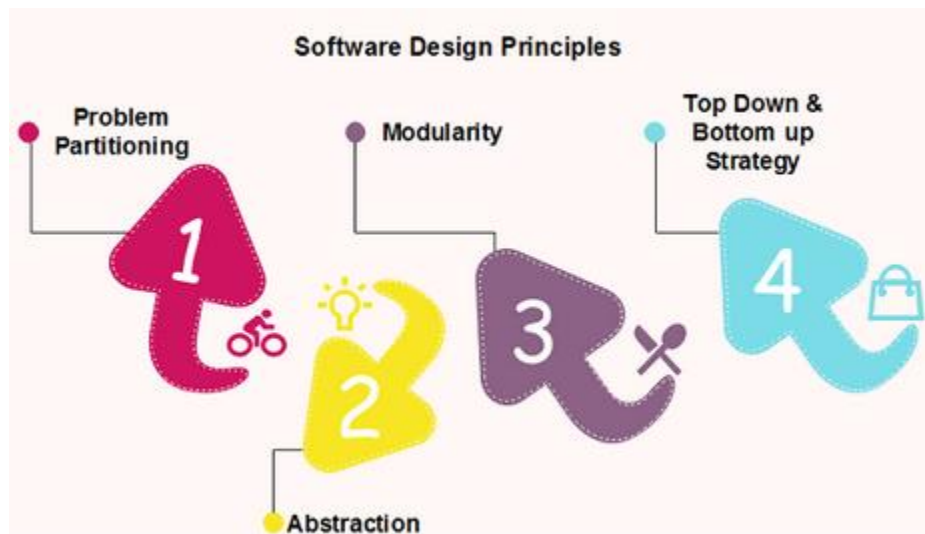
- ✓ **Software Design** is also a process to plan or convert the software requirements into a step that are needed to be carried out to develop a software system.
- ✓ There are several principles that are used to organize and arrange the structural components of Software design.
- ✓ Software Designs in which these principles are applied affect the content and the working process of the software from the beginning.



# Cont...

- Software design principles are concerned with providing means to handle the **complexity of the design process** effectively.
- Effectively managing the complexity will not only reduce the effort needed for design but can also reduce the scope of introducing errors during design.

## Following are the principles of Software Design



# Problem Partitioning

- ✓ divide the problems and conquer the problem it means to divide the problem into smaller pieces so that each piece can be captured separately.
- ✓ For software design, the goal is to divide the problem into manageable pieces.
- ✓ breaking down a complex problem or task into smaller, more manageable sub-problems or tasks. It is a top-down approach
- Benefits of Problem Partitioning
  - Software is easy to understand
  - Software becomes simple
  - Software is easy to test
  - Software is easy to modify
  - Software is easy to maintain
  - Software is easy to expand

# Abstraction

- ✓ An abstraction is a tool that enables a designer to consider a component at an abstract level without bothering about the internal details of the implementation.
- ✓ Abstraction can be used for existing element as well as the component being designed.
- ✓ Abstraction is the selection of a set of concepts to represent a complex whole.
- ✓ Abstraction is mostly formed by reducing information content of a concept or a physical phenomenon typically to retain information that is relevant to a particular purpose
- ✓ There are two common abstraction mechanisms
  - Functional Abstraction- forms the basis for Function oriented design approaches.
  - Data Abstraction- forms the basis for Object Oriented design approaches.

# Modularity

- ❑ Partitioning a problem in to sub parts (Modules).
- ❑ Modularity specifies to the division of software into separate modules which are differently named and addressed and are integrated later on in to obtain the completely functional software. It is a bottom-up approach to system design
- ✓ Quality attributes:
  - Well-defined: Modules are clearly distinguished
  - Each module has single specified objectives
  - Loosely coupled- It measures the relative function strength of a module.
  - Cohesive- It measures the relative interdependence among modules.
- ❑ A good design is the one that has low coupling.
- ❑ Coupling is measured by the number of relations between the modules.
- ❑ That is, the coupling increases as the number of calls between modules increase or the amount of shared data is large. Thus, it can be said that a design with high coupling will have more errors.

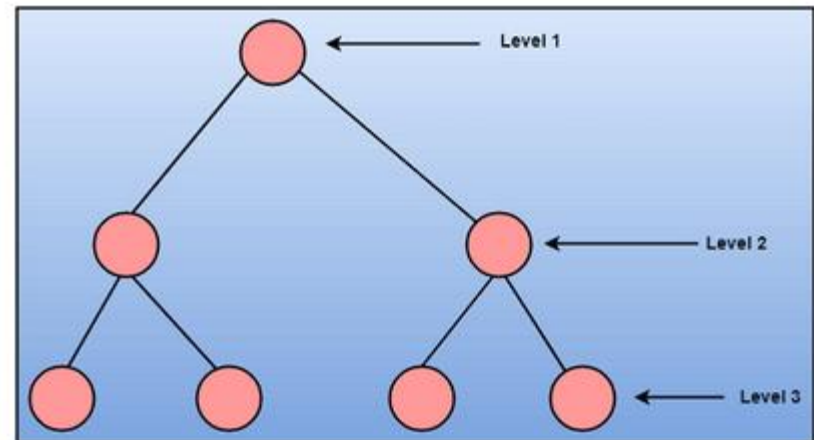
# Cont...

## ✓ Advantages of Modularity

- It allows large programs to be written by several or different people
- It encourages the creation of commonly used routines to be placed in the library and used by other programs.
- It simplifies the overlay procedure of loading a large program into main storage.
- It provides more checkpoints to measure progress.
- It provides a framework for complete testing, more accessible to test
- It produced the well designed and more readable program.

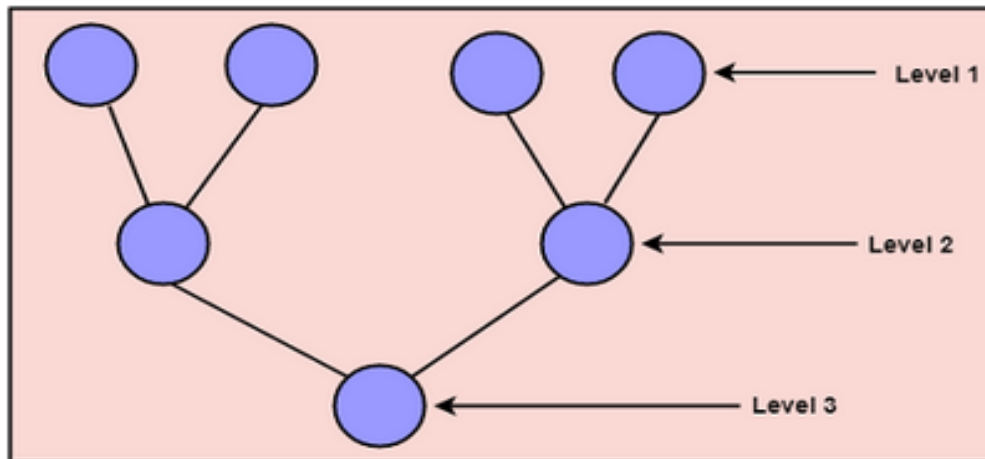
# Strategy of Design

- ✓ To design a system, there are two possible approaches
  - Top-down Approach
  - Bottom-up Approach
- ✓ **Top-down Approach:** This approach starts with the identification of the main components and then decomposing them into their more detailed sub-components.



## Cont...

- ✓ **Bottom-up Approach:** A bottom-up approach begins with the lower details and moves towards up the hierarchy, as shown in fig. This approach is suitable in case of an existing system.



# 1.6. Introduction to User Interface Design

- ✓ UI design refers to the visual design of a digital product's interface.
- ✓ It's the process of creating interfaces (namely apps and websites) with a focus on look, style, and interactivity.
- ✓ A UI designer will design the movement between different screens, and create the visual elements and their interactive properties that facilitate user interaction.
- ✓ While UI design is largely a visual discipline, UI designers work on a vast array of projects.
- ✓ In order to design accessible, user-friendly interfaces, UI designers bring empathy into every stage of the design process from designing screens, icons, and interactions, to creating a style guide that ensures consistency and the proper implementation of a visual language across an entire product.
- ✓ In today's digital world, UI designers are needed across all industries from banking, healthcare, and education to ecommerce, fashion, and travel.



## 1.6. Introduction to User Interface Design

- ✓ **What's the difference between UI design and UX design?**
- UX designers enhance user satisfaction by improving the usability and accessibility of a product,
- while UI designers enhance user satisfaction by making the product's interface look and feel enjoyable for the user.
- UX designers draw out the map of the product, which UI designers then flesh it out with visual and interactive touchpoints.
- UX and UI designers often work in tandem to maximize the responsiveness, efficiency, and accessibility of a website.

## 1.6. Introduction to User Interface Design

- AI tools can help UI designers with preliminary research, image and text generation, and other shortcuts to guide their design process.
- However, it's important to know that AI cannot replace designers completely. The knowledge a human designer brings, like user research, visual design, user behavior, and company preference is still necessary for creating efficient designs.