

1.List out any four design principles

Balance: Balance in design refers to the distribution of visual elements within a composition. It can be achieved through symmetry or asymmetry . Balance ensures that no single element overpowers the others and creates a sense of equilibrium in the design.

Contrast: Contrast involves using differences in elements like color, size, shape, or texture to create visual interest and emphasize important parts of a design. It helps guide the viewer's attention and can make a design more engaging and dynamic.

Hierarchy: Hierarchy is about organizing design elements to convey a clear order of importance or significance. It helps users or viewers understand the content or message by prioritizing and structuring information in a visually logical manner.

Alignment: Alignment is the arrangement of elements along a common axis or edge. Proper alignment creates a sense of order and cohesion in a design, making it easier to read, understand, and navigate.

2.Identify the various levels of design model.

Strategic Level

Conceptual Level

Structural Level

Functional Level

Aesthetic Level

Detail Level

Prototyping and Testing Level

Production Level

Maintenance and Evaluation Level

3.Distinguish between Cohesion and Coupling.

Cohesion	Coupling
Cohesion is the concept of intra-module	Coupling is the concept of inter-module.
Cohesion represents the relationship within a module.	Coupling represents the relationships between modules.
Increasing cohesion is good for software.	Increasing coupling is avoided for software.
Cohesion represents the functional strength of modules.	Coupling represents the independence among modules.
Cohesion is created between the same module.	Coupling is created between two different modules.

4.Mention the characteristics of good design.

User-Centered: Good design prioritizes the needs, preferences, and behaviors of the end-users. It aims to create a positive and seamless user experience, ensuring that the design is intuitive and user-friendly.

Clear Purpose: A well-designed product or system has a clear and well-defined purpose. It serves its intended function efficiently and effectively, without unnecessary complexity or features.

Simplicity: Good design embraces simplicity. It conveys its message or function in a straightforward and uncomplicated manner, reducing clutter and avoiding unnecessary embellishments.

Aesthetic Appeal: Aesthetics matter in design. Good design is visually pleasing, harmonious, and aesthetically satisfying. It considers elements such as color, typography, balance, and visual hierarchy.

PART-B

1. Explain the various kinds of design concepts.

Balance: Balance is a design concept that deals with the distribution of visual elements within a composition. It seeks to create a sense of equilibrium and harmony in the design.

There are two main types of balance:

Symmetrical Balance: Achieved when elements are mirrored or evenly distributed around a central axis.

Asymmetrical Balance: Involves an uneven distribution of elements that still achieves visual balance through careful arrangement.

Contrast: Contrast is the principle of creating visual interest by using differences in elements such as color, texture, size, or shape. It makes certain elements stand out and can help guide the viewer's attention.

Emphasis: This concept is about creating a focal point in a design to draw the viewer's attention to a specific element. It is often used to highlight the most important or central part of a composition.

Unity: Unity, or harmony, is the principle of ensuring that all elements in a design work together cohesively. It involves creating a sense of oneness and consistency in the design.

Rhythm and Repetition: Rhythm in design involves the regular recurrence of elements to create a sense of movement and pattern. Repetition of shapes, colors, or patterns can be used to establish rhythm and create visual interest.

Proportion and Scale: Proportion and scale are concepts that deal with the size and relationships between elements in a design. Properly scaled and proportioned elements can create a sense of harmony and balance.

Hierarchy: Hierarchy involves organizing elements in a way that communicates their relative importance. It helps guide the viewer's eye and prioritize information or content in the design.

Typography: Typography is a design concept that focuses on the choice of fonts, font sizes, spacing, and layout of text within a design. Typography can greatly impact the readability and visual appeal of textual content.

Color Theory: Color theory is a fundamental concept in design that deals with the use of color to create emotion, convey meaning, and establish visual harmony. Understanding color relationships and psychology is crucial for effective design.

Negative Space: Negative space, also known as white space, is the area around and between design elements. It plays a critical role in creating balance and readability in a design.

Grid Systems: Grid systems provide a structured framework for organizing content in print and web design. They ensure consistent alignment, spacing, and readability.

Material Selection: In industrial design, the choice of materials and their suitability for a product's function and aesthetics is a central concept.

Responsive Design: Responsive design is the concept of creating websites that adapt and respond to different screen sizes and devices, ensuring a consistent user experience.

Sustainability: In architectural and environmental design, the concept of sustainability emphasizes eco-friendly practices and materials to reduce environmental impact.

User-Centered Design: User-centered design is a concept focused on designing products, interfaces, and experiences with the user's needs and preferences at the forefront.

2. Illustrate the types of cohesion and coupling

Types of Cohesion:

Functional Cohesion:

Description: All elements within a module perform a single, specific function or task.

Example: A utility module that contains functions for mathematical calculations, such as addition, subtraction, multiplication, and division.

Sequential Cohesion:

Description: Elements within a module are related and executed in a sequence.

Example: A module that processes customer orders, which involves a series of steps, such as order validation, payment processing, and order confirmation.

Communicational Cohesion:

Description: Elements within a module share data or results.

Example: A module that calculates statistics from data provided as input and returns the results to the calling module.

Procedural Cohesion:

Description: Elements within a module are organized by a common procedure or routine.

Example: A module that manages user authentication and includes functions for login, logout, and password reset.

Temporal Cohesion:

Description: Elements within a module are related by time or sequence of execution.

Example: A module responsible for managing a game, with functions for initializing, playing, and ending the game.

Types of Coupling:

Low Coupling:

Description: Modules have minimal dependencies on each other, and changes in one module do not significantly affect others.

Example: A system where modules communicate through well-defined, loosely coupled interfaces, like APIs.

Message Coupling:

Description: Modules communicate by passing messages, with limited knowledge of each other's internal details.

Example: A system that uses message queues for communication, with each module processing messages independently.

Data Coupling:

Description: Modules share data but do not need to understand each other's logic.

Example: Modules that interact through a shared data structure (e.g., a database), with each module reading and writing data as needed.

Control Coupling:

Description: Modules share control information, potentially affecting each other's behavior.

Example: A main controller module that sends commands to subordinate modules to perform specific actions.

Content Coupling:

Description: Modules have direct access to each other's internal data and logic, which is the tightest form of coupling.

Example: Two modules that access and modify each other's private variables and functions, creating a highly interdependent and fragile system.