

### **Activities/ Assessments**

#### **1. Define the software design and discuss its importance as a software engineer:**

Software design is the process of defining the architecture, components, modules, interfaces, and other characteristics of a software system to meet the requirements of the stakeholders. It involves creating a high-level abstraction of the system that can be translated into a low-level implementation by software developers. The software design process is an essential step in the software development life cycle as it provides a blueprint for the construction of the software system.

### **Software Design Levels**

Software design yields three levels of results:

- **Architectural Design** – The architectural design is the highest abstract version of the system. It identifies the software as a system with many components interacting with each other. At this level, the designers get the idea of proposed solution domain.
- **High-level Design** – The high-level design breaks the ‘single entity- multiple component’ concept of architectural design into less-abstracted view of sub-systems and modules and depicts their interaction with each other. High-level design focuses on how the system along with all of its components can be implemented in forms of modules. It recognizes modular structure of each sub-system and their relation and interaction among each other.
- **Detailed Design** – Detailed design deals with the implementation part of what is seen as a system and its sub-systems in the previous two designs. It is more detailed towards modules and their implementations. It defines logical structure of each module and their interfaces to communicate with other modules.

The importance of software design cannot be overstated as it plays a crucial role in the development of high-quality software. Here are some of the key benefits of software design:

- **Provides a roadmap for development:** The software design provides a detailed plan for the development team to follow, which helps ensure that the software is developed according to the requirements and specifications of the stakeholders.

- Enables early detection of defects: The design process allows for the early identification of potential defects, which can be addressed before they become more costly to fix later in the development cycle.
- Enhances code maintainability: A well-designed software system is easier to maintain as the design provides a clear understanding of the system's structure and functionality. This makes it easier for developers to add new features, fix bugs, and update the software.
- Promotes reusability: A modular and well-organized design promotes code reusability, which can save time and resources in future software development projects.

## **2. Discuss the Advantage of using the modularization technique in developing your software/application:**

Modularization is the process of breaking a software system down into smaller, more manageable modules or components. Each module is designed to perform a specific function and can be tested and maintained independently. Here are some advantages of using modularization technique in software development:

- Encourages code reuse: Modularization promotes code reuse, which can save time and resources in future software development projects. Once a module is created, it can be used in other parts of the software system, reducing the amount of time and effort required to create new features.
- Enhances maintainability: By breaking the software system down into smaller modules, it becomes easier to maintain and update the system. Developers can work on individual modules without affecting the rest of the system, which reduces the risk of introducing errors.
- Increases scalability: A modular design enables the software system to be scaled more easily. New modules can be added as needed to accommodate additional features and functionality, without requiring major changes to the existing system.
- Simplifies testing: By breaking the software system down into smaller, more manageable modules, it becomes easier to test each module individually. This reduces the complexity of the testing process and makes it easier to identify and fix defects.

Overall, using modularization in software development can lead to a more maintainable, scalable, and reusable software system.

**Group Act: With your group, apply the modularization technique in your proposed system.**

Here's an example of how modularization technique we applied to an online food ordering system:

- **Identify the main functions of the system:** In an online food ordering system, the main functions might include order placement, payment processing, menu management, user account management, and order tracking.
- **Break down the functions into smaller modules:** For order placement, we could break it down into modules such as selecting items from a menu, customizing the order, and choosing a delivery or pickup option. For payment processing, we could break it down into modules such as payment gateway integration and transaction handling. For menu management, we could break it down into modules such as adding and editing menu items, setting prices, and categorizing items. For user account management, we could break it down into modules such as account creation and authentication, profile management, and order history. For order tracking, we could break it down into modules such as real-time order status updates, delivery tracking, and notifications.
- **Define the interfaces between the modules:** Each module should have a clearly defined interface that outlines how it interacts with other modules in the system. For example, the payment processing module should be designed to interface with the order placement module and the user account management module.
- **Develop and test each module independently:** Each module can be developed and tested independently of the other modules. This makes it easier to identify and fix defects, and also enables code reusability in future projects.
- **Integrate the modules into the final system:** Once all the modules have been developed and tested, they can be integrated into the final system. The interfaces between the modules should be thoroughly tested to ensure that they work as expected.

By using modularization, we have broken down the complex online food ordering system into smaller, more manageable modules. This makes it easier to develop, test, and maintain the software system, while also promoting code reuse and scalability.