Practice Assignment

Module 1	
Sr. No.	**Questions**
1	You're introducing a new team member to a large software project. How would you explain the principle of "Separation of Concerns" to help them navigate the codebase effectively?
2	A small startup wants to develop a secure payment system. Discuss how data abstraction and data encapsulation can play pivotal roles in achieving this goal.
3	You've been asked to review the architecture of an application, and you notice data types are often mixed with their representations. Describe the potential pitfalls of this approach and the benefits of distinguishing between the two.
4	An intern is struggling with understanding the difference between interface and implementation in a piece of software. Provide a real-world analogy or scenario to help clarify this distinction.
5	A company is migrating their older systems to a new platform. They are keen on using abstract data types (ADTs). Elaborate on the benefits of ADTs in such a context.
6	You're developing a library for third-party developers. While you want them to leverage your functions, you don't want them to know the internal workings. Discuss how interface vs.

	implementation is critical in this scenario.
7	During a team meeting, a debate arises about the balance between abstraction and performance. Reflect on a situation where too much abstraction might hinder system performance.
8	A junior developer asks you why they can't just use concrete classes and ignore abstract classes or interfaces. Explain the concept and benefits of data abstraction in response.
9	You've been tasked to optimize a graphics rendering engine. Describe how "Separation of Concerns" can aid in isolating and improving specific functionalities.
10	A company is designing a new database system and is considering the use of abstract data types. Illustrate with examples how ADTs might be employed in such a context.