Misconceptions

Module-1	
Misconception 1.	**Abstraction always leads to inefficient systems because it adds extra layers.**
Correct Explanation	Although the use of abstraction may result in the introduction of more layers, it does not automatically give rise to inefficiency. Abstraction often results in enhanced system organisation and maintainability by streamlining intricate elements and enabling developers to concentrate on higher-level logic. The potential drawbacks in terms of efficiency are often counterbalanced by the corresponding benefits of enhanced production and decreased occurrence of mistakes.
Misconception 2.	**'Separation of Concerns' means that every individual feature or function must be isolated from all others.**

Correct Explanation	The concept of 'Separation of Concerns' pertains to the structural organisationof a system in order to facilitate the separate handling of various capabilities or concerns. It is not imperative that each minute aspect be segregated; instead, the focus is on guaranteeing that significant, overarching capabilities or components do not excessively intermingle, hence enhancing the system's manageability and adaptability.
Misconception 3.	**Data Encapsulation is just about making all data private.**
Correct Explanation	Data encapsulation refers to the practice of combining data with the corresponding techniques that manipulate it, while also regulating the mannerin which the data is accessed. This approach ensures the secure and proper utilisation of the data. Although the process often entails the privatisation of data, the primary emphasis is on facilitating restricted access through clearly defined interfaces.
Misconception 4.	**The phrases "interface" and "implementation" can be utilisedinterchangeably.**

Correct Explanation	Although both concepts are associated with the design and use of software components, they pertain to distinct facets within this domain. An 'interface' refers to the establishment of a collection of interactions or features, but abstaining from providing explicit details about their underlying mechanisms. On the other hand, the concept of 'Implementation' pertains to the specific operational processes and methods underlying those functions.
Misconception 5.	**Abstract Data Types (ADTs) might be considered analogous to classes in the context of object-oriented programming.**
Correct Explanation	Abstract Data Types (ADTs) and classes exhibit some similarities, notably with regard to their ability to encapsulate data and functions. Nevertheless, Abstract Data Types (ADTs) are abstract descriptions that primarily emphasise the accessible operations, without providing details on their implementation. In the realm of object-oriented programming, classes possess a higher level of concreteness since they include both the definition, which is analogous to an

	interface, and the tangible implementation.