## Comprehension-Based Questions

- What purpose do arrays, tuples, and enums serve in modeling real-world data in TypeScript, and how would you choose between them?
- How do tuples enforce fixed element order and types, and can you provide an example of their use in a medical context?
- Explain how enums improve code maintainability when representing a set of related constants such as patient statuses.
- Describe how interfaces define object shapes and ensure type safety for complex objects like patient records.
- In what scenarios would you prefer classes over interfaces in TypeScript, and what benefits do classes provide?
- What are the primary differences between null and undefined, and how should each be used to represent missing or uninitialized values?
- How does TypeScript treat uninitialized variables versus explicitly null-assigned variables at runtime and compile time?
- Define a type alias and explain how aliasing complex union or tuple types can improve code readability.
- How can generic type aliases like `Container<T>` be used to wrap different data types with metadata?
- What is conditional logic in TypeScript, and how would you implement multi-way decisions such as assigning letter grades?
- Compare the use of `if…else if…else` ladders and `switch` statements for handling discrete ranges or values.
- Describe how `break` and `continue` control statements alter loop execution flow, providing code examples.
- When would you use a `do…while` loop instead of a `while` loop, and what guarantee does it provide about execution?

- Explain the differences between optional parameters ( `param?: type` ) and default parameters ( `param: type = value` ) in functions.
- How do rest parameters enable functions to accept variable numbers of arguments, and what is an example use-case?
- What are function overloads in TypeScript, and how do they allow a single function name to handle multiple call signatures?
- Describe how mapped types differ from conditional types and provide an example use of each.
- What is type narrowing, and which type guards are commonly used to refine types inside conditional blocks?
- How do declaration files ( `.d.ts` ) facilitate integration of JavaScript libraries into TypeScript projects?
- Explain the distinction between interfaces and type aliases, including when to choose one over the other.