Comprehension-Based Questions

- What is TypeScript, and how does it extend JavaScript's capabilities for large-scale application development?
- How does static typing in TypeScript help prevent runtime errors?
- Describe the rules for valid identifiers in TypeScript and give one example each of a valid and an invalid identifier.
- Explain the difference between declaring a variable with an explicit type annotation versus relying on type inference.
- How do you declare a variable that can hold any type of value, and what are the risks of using this approach?
- In what situations would you prefer `unknown` over `any`, and why?
- What is a type assertion in TypeScript, and how does it differ from a runtime type conversion?
- Define the built-in types `void`, `null`, `undefined`, and `never`, and give an example use case for each.
- How do you declare and use an array of strings in TypeScript? Provide sample syntax.
- Explain how classes and methods are defined in TypeScript, including access modifiers and constructors.
- What compiler flags would you enable to (a) disallow implicit `any` and (b) generate declaration files?
- Compare `let`, `const`, and `var` in terms of scope, hoisting, and mutability.
- Using the "bank vault" analogy, match each TypeScript built-in type (number, string, boolean, symbol) to its corresponding vault locker.
- Describe a scenario in a dynamic application where you might start with `any` and later refactor to a more specific type.