**Interface:**

An interface is a programming construct used to define the structure or blueprint of an object, class, or component. It specifies what properties, methods, or events an object must have, without dictating how they should be implemented. Interfaces are common in many programming languages, including TypeScript, Java, and C#.

1. **Defines a Contract**: An interface ensures that any class or object adhering to it implements its specified properties and methods.
2. **No Implementation**: Interfaces do not contain any logic or implementation—only declarations.
3. **Ensures Consistency**: Multiple classes or objects implementing the same interface follow a consistent structure.
4. **Supports Polymorphism**: Allows code to use different objects through a common interface.

**TypeScript code:**

interface User {

id: number;

name: string;

email: string;

isActive: boolean;

}

const newUser: User = {

id: 1,

name: "John Doe",

email: "john@example.com",

isActive: true,

};