

# Classes

In addition to [interfaces](#), Slice permits the definition of classes. Classes are like interfaces in that they can have operations and are like structures in that they can have data members. This leads to hybrid objects that can be treated as interfaces and passed by reference, or can be treated as values and passed by value.

Classes support inheritance and are therefore polymorphic: at run time, you can pass a class instance to an operation as long as the actual class type is derived from the formal parameter type in the operation's signature. This also permits classes to be used as type-safe unions, similarly to Pascal's discriminated variant records.

## Topics

- [Simple Classes](#)
- [Class Inheritance](#)
- [Class Inheritance Semantics](#)
- [Classes as Unions](#)
- [Self-Referential Classes](#)
- [Classes Versus Structures](#)
- [Classes with Operations](#)
- [Architectural Implications of Classes](#)
- [Classes Implementing Interfaces](#)
- [Class Inheritance Limitations](#)
- [Pass-by-Value Versus Pass-by-Reference](#)
- [Passing Interfaces by Value](#)