Module 4: Interfaces for Abstraction
Topic 1.2: Interfaces

Interfaces

- Set of **method signatures**
  - Name, parameters, return values
  - Implementation is NOT defined
- Used to express conceptual similarity between types
- Example: **Shape2D interface**
- All 2D shapes must have `Area()` and `Perimeter()`

Satisfying an Interface

- Type **satisfies an interface** if type defines all methods specified in the interface
  - Same method signatures
- **Rectangle** and **Triangle** types satisfy the **Shape2D** interface
  - Must have `Area()` and `Perimeter()` methods
  - Additional methods are OK
- Similar to inheritance with overriding

Defining an Interface Type

```
type Shape2D interface {
   Area() float64
   Perimeter() float64
}
type Triangle {…}
func (t Triangle) Area() float64 {…}
func (t Triangle) Perimeter() float64 {…}
```

- Triangle type satisfies the Shape2D interface
- No need to state it explicitly