Interfaces

Saber Mesgari

Generic

- Generic programming is a style of computer programming in which algorithms are written in terms of types **to-be-specified-later** that are then instantiated when needed for specific types provided as parameters.

Interfaces

- Interface is two thing:
  - It it is a set of methods
  - It is also a type

- An interface type is defined as a set of method signatures.

- A value of interface type can hold any value that implements those methods.

The Empty Interface

- An interface type is defined as a set of method signatures.
- A value of interface type can hold any value that implements those methods.

Interface Values

- Can be thought of as a tuple of a value and a concrete type

```
(value, type)
```

Type Assertion

- A type assertion provides access to an interface value's underlying concrete value.

```
t, ok := i.(T)
```

Type Switches

```
switch v := i.(type) {
case int:
    // here v has type int
case float:
    // here v has type float
default:
    // no match; here v has the same type as i
}
```