Delegates in Swift — Notes

What Is a Delegate?

A delegate is a class that acts on behalf of, or in coordination with, another class.
It's a design pattern used heavily in Swift to allow one class to communicate back to another object in a non inheritance way.

- Lets one class "delegate" responsibilities to another
- Organizes code, improving readability
- Often used for callbacks, event handling, and communication between view controllers

Create the delegate protocol to use

```
protocol ExampleDelegate: AnyObject {

   func example()
   }
```

Inside the object that needs to send updates:

```
class ExampleClass {

   weak var delegate: ExampleDelegate?

   func example()  {
            //give functionality to this delegated function

delegate?.example() // calls the function if it exists
   }
}
```

This is the class that needs to conform to the delegate

```
class ExampleUser: ExampleDelegate { // conform to protocol
   let object = ExampleClass()
   init() {
      object.delegate = self  // Assign delegate
   }

   func example() {
      // Implementation of delegated behavior
      print("Delegate method called.")
   }
}
```

Key Rules to Remember

- Use weak for delegate properties unless it must be strong (rare).
- Delegates usually live longer than the object delegating work.
- Delegates are preferred for continuous or stateful communication; closures are good for one-off callbacks.