



# Chapter 12/15

*Structures*

*Enumerations (Enums)*

*Error Handling*

*Unit Testing and XCTest Framework*

## Structures

- structure instances are **value types**
- **memberwise-initializer**
  - provided by Swift if there are no initializers

```
struct Shape {  
    var length: Int  
    var width: Int  
    var area: Int {  
        return length * width  
    }  
    // no init()  
}  
  
var shape = Shape(length: 5, width: 3) // arguments must be in-order
```

- does not support
  1. **inheritance**
  2. **sub-classing**
  3. **de-initializers**

```
struct MyStruct {  
    var name: String  
  
    init(name: String) {
```

```

        self.name = name
    }

    func message() {
        "Hello " + name
    }
}

```

## Enumerations (Enums)

- custom data types of pre-defined sets of values
- **Assigning (as a variable or argument)**

- `Enum.value`

- `.value`

- **Compiler uses *type-inference* (must be explicit and no ambiguities)**

```

enum Temperature {
    case hot
    case warm
    case cold(centigrade: Int)
}

func displayTempInfo(temp: Temperature) {
    switch temp {
        case .hot:
            print("It is hot")
        case .warm:
            print("It is warm")
        case .cold(let centigrade) where centigrade <= 0:
            print("Ice warning: \(centigrade) degrees.")
        case .cold:
            print("It is cold but not freezing.")
    }
}

displayTempInfo(temp: .cold(centigrade: -10))

```

## Error Handling

### 1. Error Enumeration

## 2. Exception Throwing Function

- **Goal:** isolate the function's specification

## 3. Exception Handler (do-try-catch)

```
// 1. Error Enumeration
enum FileTransferError: Error {
    case NoConnection
    case LowBandwidth
    case FileNotFound
}

let connectionOK = true
let connectionSpeed = 30.00
let fileFound = false

// Exception Throwing Function
func transferFile() throws -> Void {
    guard connectionOK else {
        throw FileTransferError.NoConnection
    }
    guard connectionSpeed >= 30 else {
        throw FileTransferError.LowBandwidth
    }
    guard fileFound else {
        throw FileTransferError.FileNotFound
    }
}

// Exception Handler
func sendFile() -> String {
    do {
        try transferFile()
    } catch FileTransferError.NoConnection, FileTransferError.LowBandwidth {
        return("Connection Problem")
    } catch FileTransferError.FileNotFound {
        return("File Not Found")
    } catch {
        return("Unknown Error")
    }
    return("Successful Transfer")
}

sendFile()
```

# Unit Testing and XCTest Framework

- making sure the code adheres to a function's specification (***unit***)