Intermediate Swift

Session 102

Mahyad

Error Handling

In Swift, errors are represented by values of types that conform to the “Error” protocol. This empty protocol indicates that a type can be used for error handling.

Error Handling

```
enum VendingMachineError: Error {
    case invalidSelection
    case insufficientFunds(coinsNeeded: Int)
    case outOfStock
    }
```

Error Handling

```
throw VendingMachineError.insufficientFunds(coinsNeeded: 5)
```

Error Handling

```
Types of Error Handling:
        - propagate the error from a function
        - Use do-catch block
        - Assert that the error will not occur
```

Error Handling

`Propagating Errors Using Throwing Functions`

Error Handling

Propagating Errors Using Throwing Functions

```
func canThrowErrors() throws -> String
```

Error Handling

Propagating Errors Using Throwing Functions

```
func vend(itemNamed name: String) throws {
    guard let item = inventory[name] else {
      throw VendingMachineError.invalidSelection
      }
```

Error Handling

Do-Catch

You use a do-catch statement to handle errors by running a block of code. If an error is thrown by the code in the do clause, it is matched against the catch clauses to determine which one of them can handle the error

Error Handling

Do-Catch

```
do {
  try buyFavoriteSnack(…)
  } catch VendingMachineError.invalidSelection {
    print(“Invalid Selection”)
    }
```


KCL TECH SOCIETY