Advanced Persistence
CMSC 436
Core Data

*Core Data* is a database

- Object-relational model (object graph)
- On disk, usually backed by SQLite
- Supports set and fetch
- Handles serialization/deserialization for you
- Very performant

Easiest way to add Core Data to your app — select checkbox at app creation

See
for a good overview
Components of Core Data

xcdatamodeld  Where you declare your core data model
  ▶  Object types are *Entities*
  ▶  Auto-generates code
  ▶  Can generate swift files

Persistence.swift  Defines a PersistenceController
  ▶  Singleton instance `.shared`
  ▶  Creates a container for objects
  ▶  Container has a `viewContext` for in-memory cache
  ▶  Lets you create a special instance for preview

Wiring into app  Set `viewContext` as
  \`.managedObjectContext` in environment

Using in view  Bind `\`.managedObjectContext` from environment
Data Model

Define your *Entities* in the `xcdatamodeld` editor:

Default *Codegen* is *Class Definition*

Switch to *Manual/None* if you’re going to manually generate
import Foundation
import CoreData

@objc(Thingy)
public class Thingy: NSManagedObject {
}

extension Thingy {
    @nonobjc public class func fetchRequest()-> NSFetchRequest<Thingy> {
        return NSFetchRequest<Thingy>(entityName: "Thingy")
    }

    @NSManaged public var a: Int16
    @NSManaged public var b: String?
    @NSManaged public var c: Bool
    @NSManaged public var anotherThingy: AnotherThingy?
}

extension Thingy : Identifiable {
}
Fetch Requests

Contain:
- the entity (prior to iOS 15)
- sort descriptors
- predicates (think SQL, not regular expressions)

```swift
@FetchRequest(
    sortDescriptors: [NSSortDescriptor(keyPath: \Word.spelling, ascending: true)],
    animation: .default)
private var items: FetchedResults<Word>
```
Predicates

String comparisons:

- BEGINSWITH
- CONTAINS
- ENDSWITH
- LIKE
- MATCHES - ICU regular expressions!