1️⃣

Session #2: Take Home Assignment

📖 Required Reading

Complete these readings **before** starting the coding assignment:

- [ ] Control Flow
- [ ] Functions
- [ ] Closures
- [ ] Enumerations

💻 Coding Assignment

Deliverable: Create `Session2Assignment.swift`

Part 1: Functions

- [ ] **Task 1.1:** Rectangle Area Calculator
  - Function name: `calculateRectangleArea`
  - Parameters: width and height (with default values)
  - Return: Area as a Double
  - Test with: No parameters, one parameter, both parameters
- [ ] **Task 1.2:** Min/Max Finder
  - Function name: `findMinMax`
  - Parameter: Array of numbers
  - Return: Tuple containing (min, max)

- Handle edge case: empty array

Questions to Consider:

- Why would default parameters be useful in real iOS apps?
- When would you use tuples vs creating a custom struct?

Part 2: Closures

☐ **Task 2.1:** Filter Odd Numbers

- Create an array of integers (1-20)
- Use `filter` with a closure to keep only even numbers
- Print the result

☐ **Task 2.2:** Counter Closure

- Create a closure that captures a counter variable
- Each time the closure is called, increment and return the counter
- Demonstrate calling it multiple times

Questions to Consider:

- What does "capturing" mean in the context of closures?
- How is this different from passing a parameter?

Part 3: Collections - Transformations

☐ **Task 3.1:** Square Numbers with Map

- Create an array: `[1, 2, 3, 4, 5]`
- Use `map` to create new array with squared values
- Print both arrays

☐ **Task 3.2:** Shopping Cart Total with Reduce

- Create an array of item prices (e.g., `[29.99, 15.50, 8.99, 42.00]`)
- Use `reduce` to calculate the total

- Print formatted total with currency symbol

Domain Modeling Insight: *Think about how these operations model real-world transformations in apps*

Part 4: Collections - Sets & Dictionaries

☐ **Task 4.1:** Common Friends Finder

- Create two Sets representing friend groups:
  - `friendsGroupA`: Set of names
  - `friendsGroupB`: Set of names
- Find common friends using set operations
- Print the result

☐ **Task 4.2:** Student Grades Manager

- Create a dictionary: `[String: Int]` (student name: grade)
- Add at least 5 students with grades
- Update one student's grade
- Filter students with grades above 80
- Print the filtered results

Questions to Consider:

- Why use a Set instead of Array for finding common elements?
- What are the performance implications?

Part 5: Bonus Challenge 🌟

☐ **Performance Comparison: Array vs Set**

- Create a large Array (10,000+ elements)
- Create a Set with the same elements
- Measure time to search for an element in both
- Use `CFAbsoluteTimeGetCurrent()` for timing

- Document your findings

Expected Investigation:

- Why is Set faster for lookups?
- When would you choose Array over Set despite performance?

📚 Additional Resources

Optional materials for deeper learning:

- Swift Standard Library - Collections
- Higher-Order Functions in Swift
- Measuring execution speed using CFAbsoluteTimeGetCurrent()