🟢 Beginner Level (Very Very Easy)

👉 Focus: Java 8+ feature basics, no logic twist

✅ Assignment 1: Simple Name Sorter

Take 5 names and sort them using a lambda expression.

📦 Feature: Lambda Expressions

👉 Hint: Use comparator with lambda.

✅ Assignment 2: Safe Divider

Divide 2 numbers. If divisor is 0, return “Not Allowed” using Optional.

📦 Feature: Optional

👉 Hint: Use Optional.empty() for 0 case.

✅ Assignment 3: Greetings Printer

Pass a message printer function to another method using functional interface.

📦 Feature: Functional Interface

👉 Hint: Use interface with 1 method.

✅ Assignment 4: Default Power

Create an interface with one abstract method and one default method. Call both.

📦 Feature: Default Methods in Interface

👉 Hint: Use default inside interface.

🟡 Intermediate Level (Very Easy, Slightly Twisted)

👉 Focus: Functional programming + basic stream use

✅ Assignment 5: A-Name Finder

Filter a list of names and keep only those starting with "A".

📦 Feature: Stream Filter

👉 Hint: Use stream with filter().

✅ Assignment 6: Word Counter

Count how many names are longer than 5 characters.

📦 Feature: Stream Filter + Count

👉 Hint: Use .filter().count().

✅ Assignment 7: Even Number Collector

Take a list of numbers and collect only even ones.

📦 Feature: Stream + Collectors

👉 Hint: Use .filter() and .collect().

✅ Assignment 8: Pretty Joiner

Join a list of city names into one string, separated by commas.

📦 Feature: Collectors.joining()

👉 Hint: Use .collect(joining()).

🔴 Advanced Level (Easy but Real-World Structured)

👉 Focus: Functional interfaces + advanced streams + multithreading

---

✅ Assignment 9: Parallel Square Printer

Print squares of numbers using parallel stream.

📦 Feature: parallelStream

👉 Hint: Use .parallelStream().forEach()

✅ Assignment 10: Frequency Mapper

Given a list with duplicate names, create a map with each name’s count.

📦 Feature: Stream + GroupingBy

👉 Hint: Use .groupingBy(..., counting())

✅ Assignment 11: Custom Filter Chain

Create 2 predicates (starts with A, ends with a) and chain them to filter names.

📦 Feature: Predicate Interface

👉 Hint: Use .and() or .or() with predicates.

✅ Assignment 12: Method Reference Sorter

Sort a list of strings using method reference instead of lambda.

📦 Feature: Method Reference

👉 Hint: Use Class::instanceMethod