Collections

1. Write a function that takes a list of integers and returns a new list with the elements in reverse order.
2. Write a function that takes a list of integers and an integer n, and returns a new list with all elements less than n removed.
3. Write a function that takes a list of strings and returns a map where the keys are the strings and the values are their lengths.
4. Write a function that takes two lists of integers and returns true if the first list is a subset of the second list.
5. Given a list of Person objects (each with name and age properties), sort the list first by age in ascending order and then by name in alphabetical order if ages are equal.

Null Safety

1. Safe Calls and Elvis Operator

Write a function that takes a nullable string and returns its length if it is not null, or -1 if it is null. Use the safe call operator (?.) and the Elvis operator (?:).

2. Nullability and Collections

Given a list of nullable integers, write a function that returns a list of non-null integers. Use the filterNotNull function.

3. Handling Null with let

Write a function that takes a nullable string and prints its uppercase version if it is not null, otherwise prints “String is null”. Use the let function.

4. Null Safety in Data Classes

Define a data class User with nullable properties name and email. Write a function that prints “Incomplete User” if any property is null, otherwise prints the user’s details.

5. Safe Casting with as?

Given a list of Any type, write a function that filters out all elements that are not strings and returns a list of strings. Use safe casting (as?).