Scala

Functional programming

What Is Functional Programming?

1- Programming paradigm

2- uses functions as the central building block of programs

3- use pure functions and immutable values

Immutability

value or state of variables, objects can’t be changed.

we can create a new object, but we cannot modify the existing object’s state.

- **Immutable objects are more thread-safe than mutable objects.**
- **Case classes are good for modeling Immutable Data**

Functions as First-Class Citizens

When we treat a function as a value

a first-class function can be:

- **Assigned to a variable**
- **passed as an argument to other functions**
- **returned as a value from other functions**
- **Higher-order functions are a key characteristic of first-class citizenship**

Scala treats all functions as first-class functions by default.

Higher-Order Functions (HOF)

- **Take other functions as arguments**
- **And/Or return functions as results**
- **Or both**

It’s allowing for a high degree of abstraction and code reuse

Functional Transformations On Collections

Operations that manipulate the elements of a collection using higher-order functions like:

- **map : Transform each element of a collection**
- **filter : Filter elements based on predicate**
- **flatMap : Transform and flatten nested collections**
- **foldLeft : Aggregate elements from left to right**
- **groupBy: Group elements based on a key**
- **foreach: Apply a function to each element without returning a new collection.**
- **Chaining operations**