Scala Tutorial 1 – 22001956

1. It is a general-purpose, high-level, multy paradigm programming language which supports both functional programming and object-oriented programming, and is designed to address the limitations of Java while running on the Java Virtual Machine.
(Father of Scala – Martin Odersky)

2. Yes, scala is a statically-typed language.
Statically typed languages catch errors early but require explicit type declarations, while dynamically typed languages offer ease of use but perform type checks during execution.

3. Scala is not pure OOP as it allows mixing of FP and OOP.
Java is also not a pure OOP language as it contains static methods and primitive data types.

4. Both Scala and Java 8 support all FP concepts.

5. Major advantages: Need fewer lines of syntax, supports FP, statically-typed and hence strong type system, compatible with JVM, powerful IDE support.
Drawbacks: Learning curve is difficult for beginners, libraries are not mature, scala build tools are complex

6. Learning curve

7. To provide a language that combines best of both worlds: OOP and FP

8. Java, Kotlin, Scala

9. Any class

10. Public (No, it doesn't have the public keyword)

11. A feature that allows the compiler to automatically deduce the type of an expression without having to explicitly declaring it.

12. Scala's Int is a primitive data type that represents an integer.
Java's java.lang.Integer is an object wrapper for the primitive int.

13. Nothing – subtype of every other type and has no instances.
Nil – An object representing an empty list of type List[Nothing]

14. Null – Is a trait in Scala that is a subtype of all reference types, used to denote the absence of an object
null – is a singleton instance of the Null type.

Difference: Null is a type and null is the value that indicated the absence of an object.

15. Unit – is a type in Scala representing a value that is used where no meaningful value is returned.

Unit vs void – void signifies that a method does not return a value. Unit is similar but is a first-class value.

16. val – defines an immutable variable
var – defines a mutable variable

17. REPL – Read-Eval-Print Loop – is an interactive shell that reads user inputs, evaluates them, and returns the result. Used for quick experimentation and testing of Scala code snippets. Open command prompt and type scala to access REPL.

18. Statically typed, type inference, immutable collections, FP support, OOP support, concise syntax, interoperability with Java, Pattern matching, concurrency support

19. Using recursion or higher-order functions.
Difference – OOP loops use imperative constructs, FP loops use immutable data and functions

20. Application – defined by creating an object that extends the App trait.
App – a trait in Scala that provides a main method, allowing the object to be used as a runnable application. It simplifies the creation of entry points in Scala programs.

21. Scala support Operator Overloading but Java does not.

22. Expression – a piece of code that produces a value
Statement – piece of code that performs an action
Difference – expressions return values, statements perform actions and do not return values

23. Java's If...Else – a control flow statement that does not return a value
Scala's If..Else – an expression that return a value

24. Scala – expression based language
Java – statement based language

25. Java supports – checked exceptions, enum types, native methods
Scala supports – pattern matching, type inference, traits, unified type system

26. Function is a first-class value, which can be passed around as a value. Method is a part of a class or object and is invoked on an instance of that class or object.

27. Can define multiple public classes in a single Scala source file.

28. scala._
java.lang._
scala.Predef._

29. no fixed number

30. static, throws, extends, implements
Scala has a unified type system and advanced features like traits and pattern matching.

31. An object that provides a number of commonly used functions and type aliases