Scala Tutorial 01

01)Scala is a programming language. Scala is both OOP and FP.

Martin Odersky is the father of scala.

02) Yes. Scala is a statically-type language

Statically Typed language

In statically Type languages, the type of variable checking is performed during compilation, and any type errors must be corrected before the code can successfully compile and run.

Dynamically Typed language

In Dynamically type languages, the type of variable checking is done as the program is executing. This can lead to runtime errors if types are misused.

Different between Statically and Dynamically Type language

Statically Type language	Dynamically Type language
01)Type checking is done at compile time 02)Have better performance 03)Less flexible 04)Easier to debug and maintain 05)Initially slower development	01)Type checking is done at running time 02)Have slower performance 03)more flexible 04)Harder to debug and maintain 05)Initially faster development

03)No

04) Scala is designed to fully support functional programming concepts.

Java 8 introduced several features to support functional programming concepts, although it does not support all FP concepts as comprehensively as scala.

05)

Major Advantages of Scala language

1. Scala seamlessly integrates object-oriented and functional programming paradigms, allowing developers to leverage the strengths of both.

2.Scala runs on the Java Virtual Machine, which allows it to interoperate with java. This mean you can use java libraries and frameworks in Scala.

3. Scala's type system is both expressive and powerful, allowing for advanced type inference and type safety.

4. As the name suggests, Scala is designed to be scalable and can handle everything from small scripts to large systems.

Drawbacks of Scala Language

1.Scala's compilation times can be significantly slower than java's ,especially for large projects, which can impact development speed.

2.Scala's tooling is generally less mature.

3.The number of libraries specifically written in scala is smaller compared to java.

06)In Scala , complexity and the associated learning curve are often cited as its primary drawback

07) The main motto of Scala is "Scalable Language." This reflects Scala's design philosophy of being adaptable to various use cases, ranging from small scripts to large, complex applications. The language aims to be scalable in terms of both its ability to grow with the needs of the user and its capacity to integrate and work seamlessly within different computing environments and platforms.

08)Java,Kotlin,Scala,Groovy,Clojure,Frege,Fantomas,Jython,JRuby

09)Scala.Any

10)Public

Scala does not use the public keyword

11) Type inference in Scala refers to the compiler's ability to automatically deduce the data types of variables, expressions, and function return types based on the context and values used in the code. This allows Scala developers to write concise code without explicitly specifying types while ensuring type safety.

12)Int: primitive type in scala. Integer: Wrapper class in java.Int and Richlnt: ‘int’ in scala can be implicity converted to ‘Richlnt’ providing additional methods.

13) `Nothing` is a subtype of all other types in Scala. It represents a type that has no instances. It's commonly used to indicate abnormal termination, such as in the return type of a method that throws an exception.

`Nil` is an empty list in Scala. It's an instance of the `List` class with no elements. It's often used when you need an empty list, especially in pattern matching or list operations.

The relationship between `Nothing` and `Nil` is that `Nil` is an instance of `List[Nothing]`. Since `Nothing` is a subtype of all types, `List[Nothing]` is a subtype of `List[T]` for any type `T`. This means that `Nil` can be used as an empty list for any type of elements.

14)Null: Trait representing the type of references that can be assigned ‘null’. It’s rarely used directly

null: is the literal used to denote a null reference, i.e a reference that doesn’t point to any object

15) both `Unit` in Scala and `void` in Java represent the absence of a meaningful value, `Unit` is more versatile in Scala because it is an object type and can be used in more contexts than `void` in Java.

16)val: Immutable variable(like ‘final’ in Java) var: Mutable variable

17)REPL: Read -Eval-Print Loop an interactive shell for scala.Access: Run ‘Scala’ from the command prompt.

18)Statically-typed ,type inference, supports OOP and FP,Runs on JVM.

19) Implemented using recursion or higher-order functions like ‘map’, ‘filter’. ’reduce’.

20)Application: A program with a main method . App: Trait is Scala for quickly creating applications