Types of Languages

* Computers internally are ___ -(1)
* Programming Language -(2)
* Why do we need programming languages -(3)
* Types of languages -(4)
* What is Procedural language -(5)
* What is Functional language -(6)
* Does Functional Programing Language follows FCF -(7)
* What is First class Function -(8)
* Imperative is ___ -(9)
  Declarative is ___
* What is Pure function. -(10)
* What is OOP -(11)
* What is Static language -(12) What is compilation
* How does the PL know the type of variable -(13)
* what is dynamic language -(14)
* Runtime -(16)
* What is the 2 types of memory management -(6)
* Garbage collection.
* Pass by reference value.

We give instruction to computer it will give us some output.
ex: turn on the lights.

1 → Binary number

2 → * We use programming language to interact with computers.
    * Or its a instruction given to computer

4 → * procedural
    * Functional
    * Objected Oriented

5 → * series of well structured steps & procedures to write a program. It contains a systematic

order of statements, functions and commands to complete a task/program

* input A
  input B ✓ input A, add A+B, input B ✗ not procedural
  add A+B
* Java follow procedural properties

6 → * Function is a piece of code that you can reuse over an over again. in our code.
* A Function can hold one or more complete lines of code.
* If in every file take 2 nos & print sum you can write it in the main file & reuse it again in the other file
* FPL it can be used in situation where we have to perform lots of different operation on the same set of data because its not modifying the original data it is creating new one as output

7 → * F.C.F
Can be assigned to regular variables
Passed as arguments to functions
Returned as results of functions
Included in data structure

8 → * we know that variable can reassign if your able to reassign function name to variable other function as well that means it is FCF.
* a=10, b=20, c=b then c=20 it is possible if FCF

9 → * Procedure
* Functional

10 → * Pure function never modify variable (a=10 // a is variable, value of that is 10) it going to create new ones as a output.

11 → Object Oriented language
* Code + data (String, integers etc) * Classes are named group of properties & function
Car object name = wheels, Properties. * Developed to make it easier to develop, debug, reuse & maintain software
Can go to maintenance we will change particular part of it not change entire as
Likewise code is divided in section if any issues with engine code then only that particular section is changed rather than whole section
* Python, C++, Java.
* Object group together a set of variables & functions to create a model of a something you would recognize from the real world.
* In Object variable → properties
function → method This collection of all is called classes & instance of this class is called Object

12 → * Type checking will be done at compilation time
* Conversion of source code to machine code is called compilation.
* Int a= 10; int a=10 a="kumar", Error as it is not integer
* Declare datatype hence more control over the data so runtime error will be reduced

* While the program is compiling while its converting your source code to machine code during that conversion time programming language should know what is the type of a. Here type checking done at compile time int a = 10;
  * int a = 10; → Compilation → Runtime
  * → 'a' + 10 = error int a = 10 error
    string + integer = error int a = 'Kunal'

13 →

14 → * When the compilation is done & program is running that time if the PL this is integer then it is Dynamic language, where you donot have to worry about specify type previously.
* It is like I will write a variable you figure it out yourself

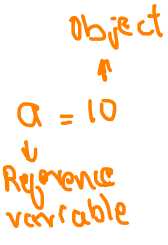
Dynamic languages
* Donot have to worry about specifically typing the types.
* Type checking at Runtime
* a = 10; a = 10 → will be removed by Garbage collected in python. a = 'Kunal' → No error first a was pointing to 10 then pointed to string outcome will be string
* No need to declare datatype., saves time

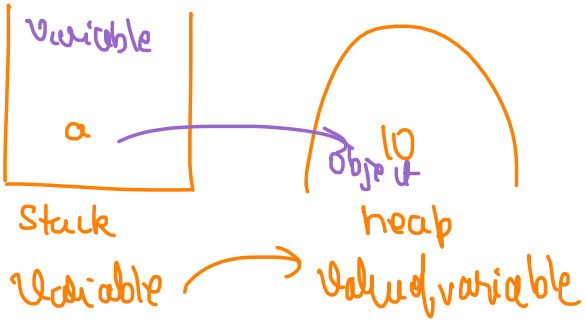
a = 10 no error
a = 'Kunal'

15 → Runtime → program is running after compilation, once the program has been converted into machine code when that machine code is running

16 → * Stack
* Heap

a will be pointing towards the object





Variable → Value of variable

* More than one reference variable can point to the same object → (1)
* If any one of these reference variables change the object / Original object is going to be changed for all the reference variable. Java (Java has only pass by reference value)

a = [1, 2, 3, 5]
b = a

a → [1, 2, 3, 5]

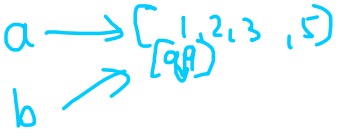
* If the object was changed

a = [1, 2, 3, 5]

b = a

a[0] = 99

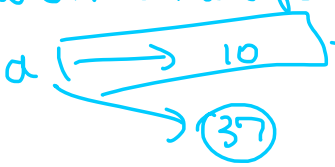
Output b = 99.



b = [99, 2, 3, 5]

* If the object was changed via 1 reference variable the change will be for all the other reference variable as well.
* Mutability (Immutable) sometimes it won't show

17 → * When there is no reference variable to the object. this will be removed when garbage collection hits



GarbageCollection

* object with no reference variable will be removed

18 → Java is object oriented language.