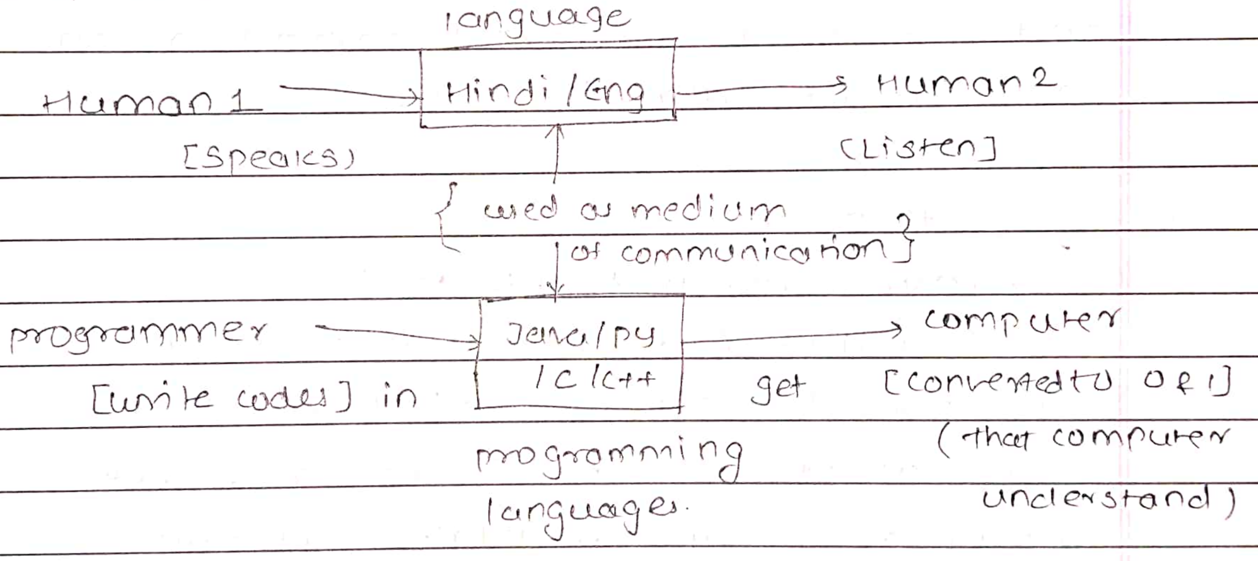
Introduction to Programming

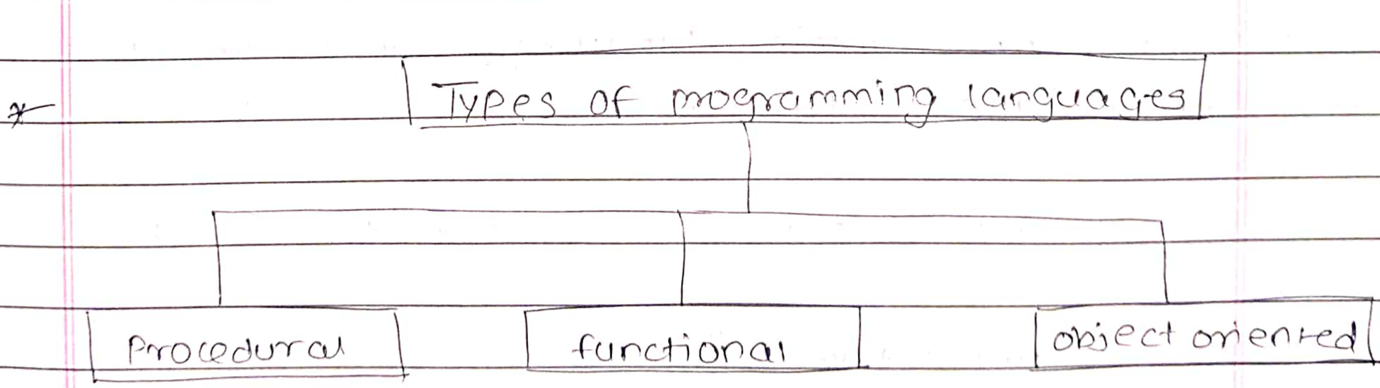
* What is Programming?

- programming is a way to instruct the computer to perform various task.
- computers only understand Binary i.e. 0's & 1's.
- Instructing computers in Binary i.e. 0's & 1's are very difficult for human. so, to solve this issue programming languages are introduced.

* what is programming language :-

It is a human readable computer language used by programmers to communicate with computers.





1) procedural :-
- specifies a series of well-structured and steps and procedures to compose a program.
- contains a systematic order of statements, functions & commands to complete a task.

2) functional :-
- writing program only in pure functions i.e., never modify variables but only create new ones as an output.
- used in a situation where we have to perform lots of different operations on the same set of data like ML.

3) object oriented :-
- Revolves around objects.
- combination (code + Data) → object.
{ instance of a class → object }
- Developed to make it easierto develop, debug, reuse and maintain software.

* " one programming language can be of all three type → python".

* "Java, c++ → procedural & object oriented".

* " C → procedural ".

*	Static language	Dynamic language
1)	perform data type checking at compile time	- perform data type checking at run time.
2)	At compile time program is converted to machine code	- At runtime program actually runs & it come after compile time.
3)	Error will show at compile time	- Error might not show until programs run.
4]	need to - Declare datatype before use	- no need to declare datatype of variable.
5]	more control	- save time in writing code but might give error at runtime.

* Memory management :-

- There are 2 types of memory stack and heap.
- when we declare a variable then the reference variable is stored in stack memory points to the object of that variable stored in heap memory.

e.g. a = 10 , a = reference variable
10 = object of ref^n variable

{ a is stored in stack memory
10 is stored in heap memory }

variable always ~~keep~~ points to its object

Imp points :

- more than one ref^n variable can points to same object
- If any changes made to the object of any reference variable that will be reflected to all others variable pointing to same object.
- If there is an object without variable then object will be destroyed by " Garbage collection ".