- **Procedural**
  → Specifies a series of well-structured steps and procedures to compose a program.
  → Contains a systematic order of statements, functions and commands to complete a task.

- **Function**
  → Writing a program only in pure function i.e never modify variables, but only create new ones as an output.
  → Used in situations where we have to perform lots of different operations on the same set of data, like ML.

- **Object Oriented**
  → Revolves around objects.
  → Code + Data = Objects.
  → Developed to make it easier to develop, debug, reused and maintain software.

⇒ **Static Language :-**

→ In static languages, the datatype can to be changed once a variable is created. This means that if we define an integer, we can only updates an integer,

its value and no other data, can be assigned to it.

→ perform type cheacking at compile time.

→ Errors will show at compile time.

→ Declare datatype before you use it.

→ More contole.

eg- int a= 10; (correct)
　　int a= "vishd" (x) //error.
　　String a= 20 // error

Dynamic Languages :-

→ In dynamic languages, the types and values are both dynamic, which means the types and value can both be changed. A variable that was preveously assigned an integer can be assigned a sbing. the type checking is done during run time.

→ perform type checking at runtime.

→ Error might not show till program is run.

→ No need to declare datatype of variables

→ Saves time in writing code but might give error at runtime.

Memory Managements

→ The JVM divides the memory into two parts (stack & Heap).

→ When we declare a variable then the reference variable stored in stack memory points to the object of that variable in heap memory.

int a=10;
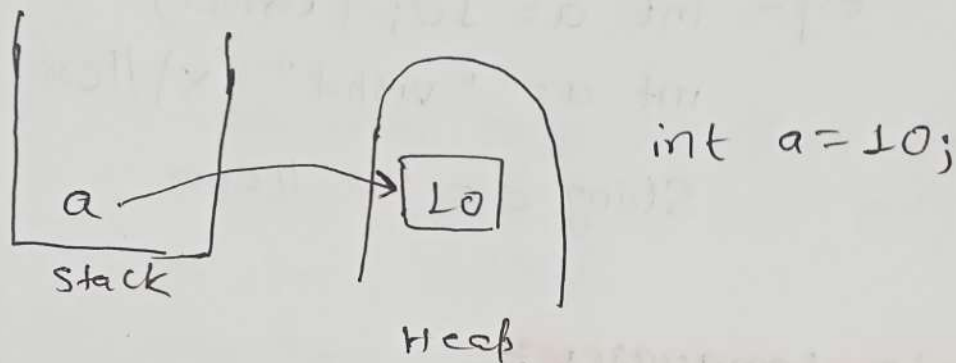
Here, a called reference variable and 10 is the object of the reference variable.

→ Reference variable are stored in stack memory.

→ Heap memory stores the objects of reference variable.

vishal
vishlu
p.d
singh
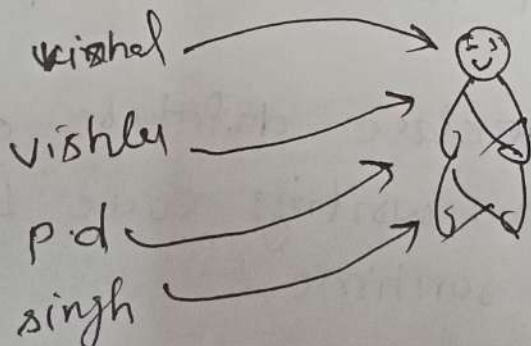
- More than one reference variable can points to same object.

- If any changes made to the object of any reference variable that will be reflected to all others variable pointing to same objects.

- If there is an object without reference variable then object will be destroyed by "Grabage collection".

eg — $a = [11, 22, 33, 44]$

      $b = a$

      $a[0] = 99$

      S.o.p (b);

         ↳ 99, 22, 33, 44

                                      99
                         a ⟶ [11, 22, 33, 44]
                         b ⟋