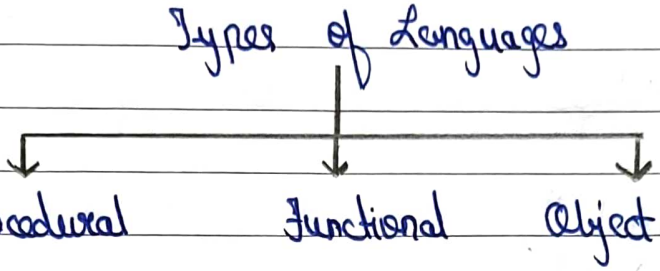
Camlin Page
Date / /

Introduction to Programming

* Types of Languages - 3 types



* Programming - Internally bunch of 0's & 1's
  - Hey computer ! add 2 numbers & display it
  ↓
  Instruction - Difficult to write in 0's & 1's
  - Human Readable format
  - C, C++, Java, Python

* Procedural

- Specifies a series of well structured steps and procedures to compose a program.
- Contains a systematic order of statements, functions and comments to complete a task.
- e.g.
  Input 1st no → Input 2nd no → Sum → Print a sum
- loops, statements, functions, commands

* Functions

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

like ML.

- Function - Bundle a code and reuse it anywhere
- e.g. a = 10;
  ↓ variable ↳ value

  name = "Kunal";
  ↓ variable ↳ value

- First class functions ?
- e.g. Python

```
a = 10;
b = 30;
c = b;
```

* Object Oriented

- Revolves around object
- Code + Data = Object
- Developed to make it easier to develop, debug, reuse, and maintain software
- e.g. Collection of all students with a single student having 3 properties.
  ↑
  data type - (int, string)
  ↓
  Custom data types with classes

* Classes - Named group of properties and functions
  ↳ collection of all - int, string, double

* Object - Instance of class

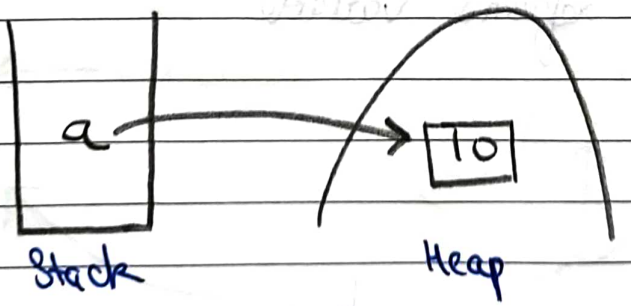
* Static & Dynamic Languages

* Static -
- Perform type checking at compile time
- Errors will show at compile time
- Declare data type before we use it
- More control

* Dynamic
- Perform type checking at compile time
- Error might not show till program is run
- No need to declare data types of variable
- Saves time in writing code but might give error at runtime.

- e.g. Static
int a = 21;
int a = "atharva"; error

a = 21 } Dynamic - no error
a = "kunal" } Static - Error

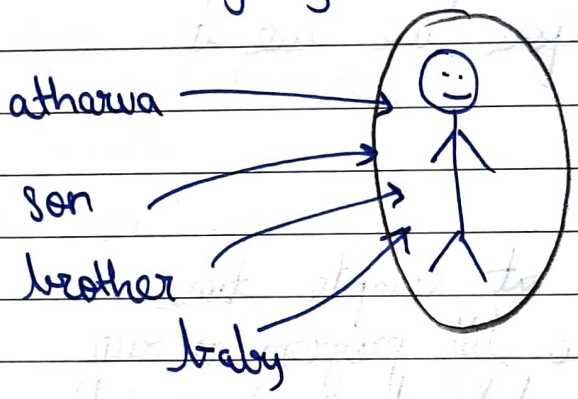


a = 10;
↓ ↳ object
reference variable

* Imp-

1 - More than 1 ref variable can point to same obj

2 - Anyone of this ref variable change the obj, original obj is going to be changed.
- & it is going to be changed for all



e.g.

a = [1, 3, 5, 9] → a → [1, 3, 5, 9]

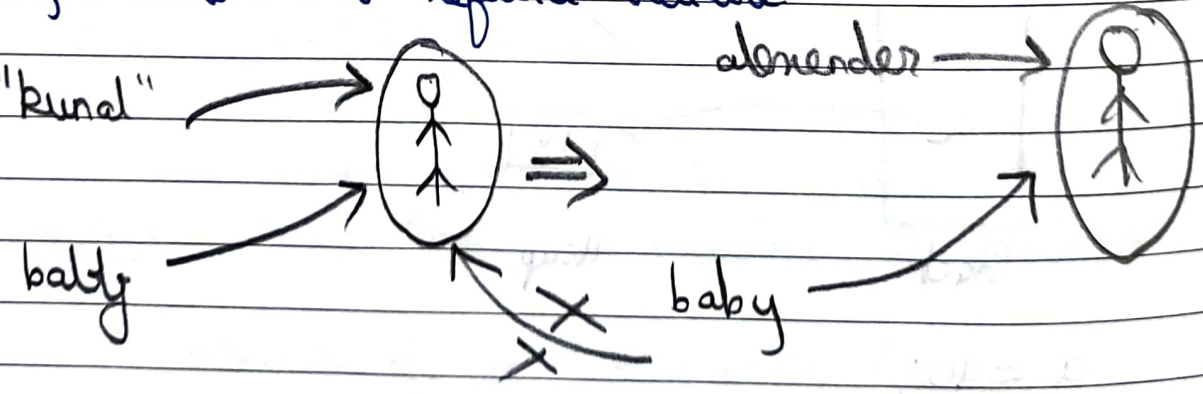
b = a → b → [1, 3, 5, 9]

a[0] = 99 → a → [99, 3, 5, 9]

Output for (b)

b = [99, 3, 5, 9]

* Object with no reference variable



→ obj with no reference variable

↓

- no variable pointing towards this object

- It will be removed from the memory when garbage collection hits.

* Garbage Collection - Obj that do not have a reference variable pointing towards them as above e.g. all the object that don't have variable pointing to them will be removed automatically.

e.g. a = 10 → a → [10]

a = 37

a → [10] → removed

a → [37]

Yamraj