Session 2
- types : any, unknown, null, void, undefined
- classes (mV, mF), object, constructor(sF)
- new, this

session01

SD → S + M

LLD =

HLD =

JS → En
OOPS

?.

this.
↓
context of
current obj

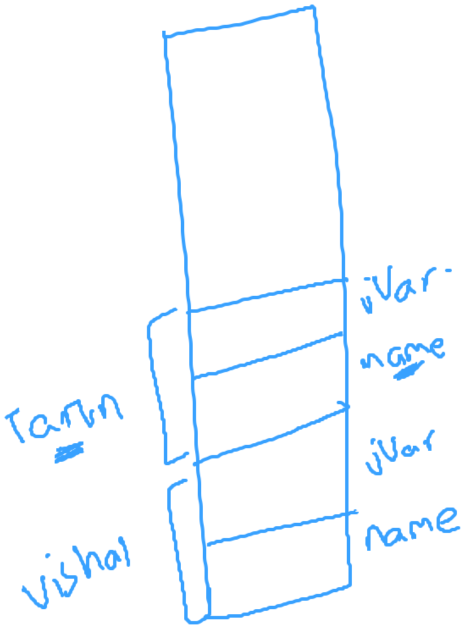
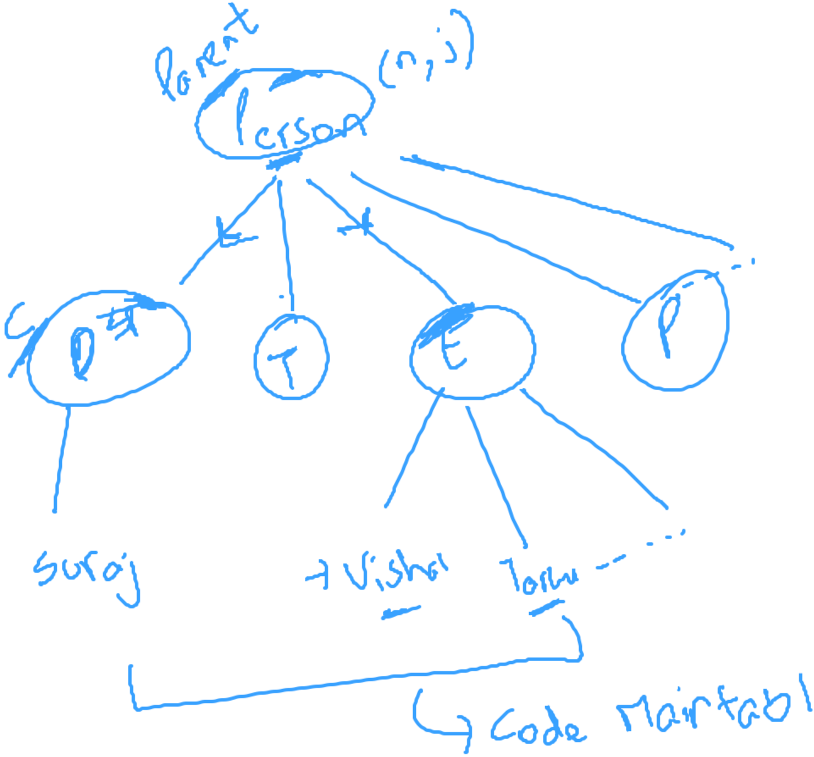


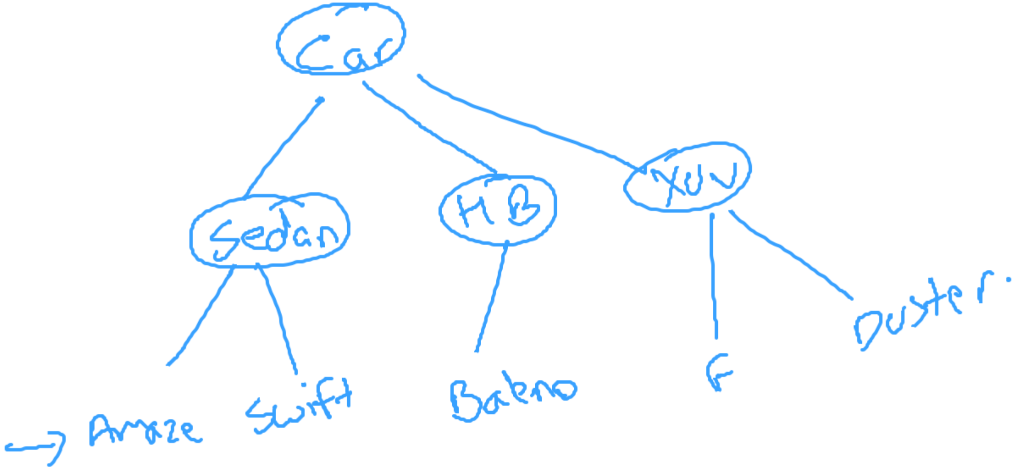
Fig: Memory Representation

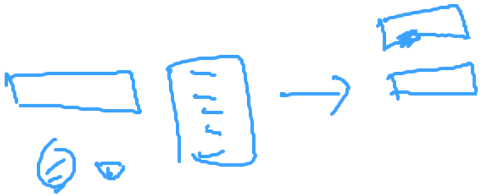
Example Class
- Person : mv(name, jobVar), mF(introduction)
  - Doctor, Teacher, Engineer, Pilot, Leader, Scientist, etc.
  - Vishal, Tarun, etc.
- Car : mV(name, type, model, color), mF(Acc, Brake, Clutch)
- Animal : mv(type, legs), mF(eat, sleep, breed)

job → Market

C







Inheritance

- Parent, Child
- inherit : mV, mF
- syntax : extends

```
class Parent{
    mv
    mF
    }

class Child extends Parent{
        mv
        mF
    }
```

- Need : Modularity, Reusability

Inheritance
+ Code Reusability
- Modularity

Interface
+ More Flexibility
- We need to write all the definitions

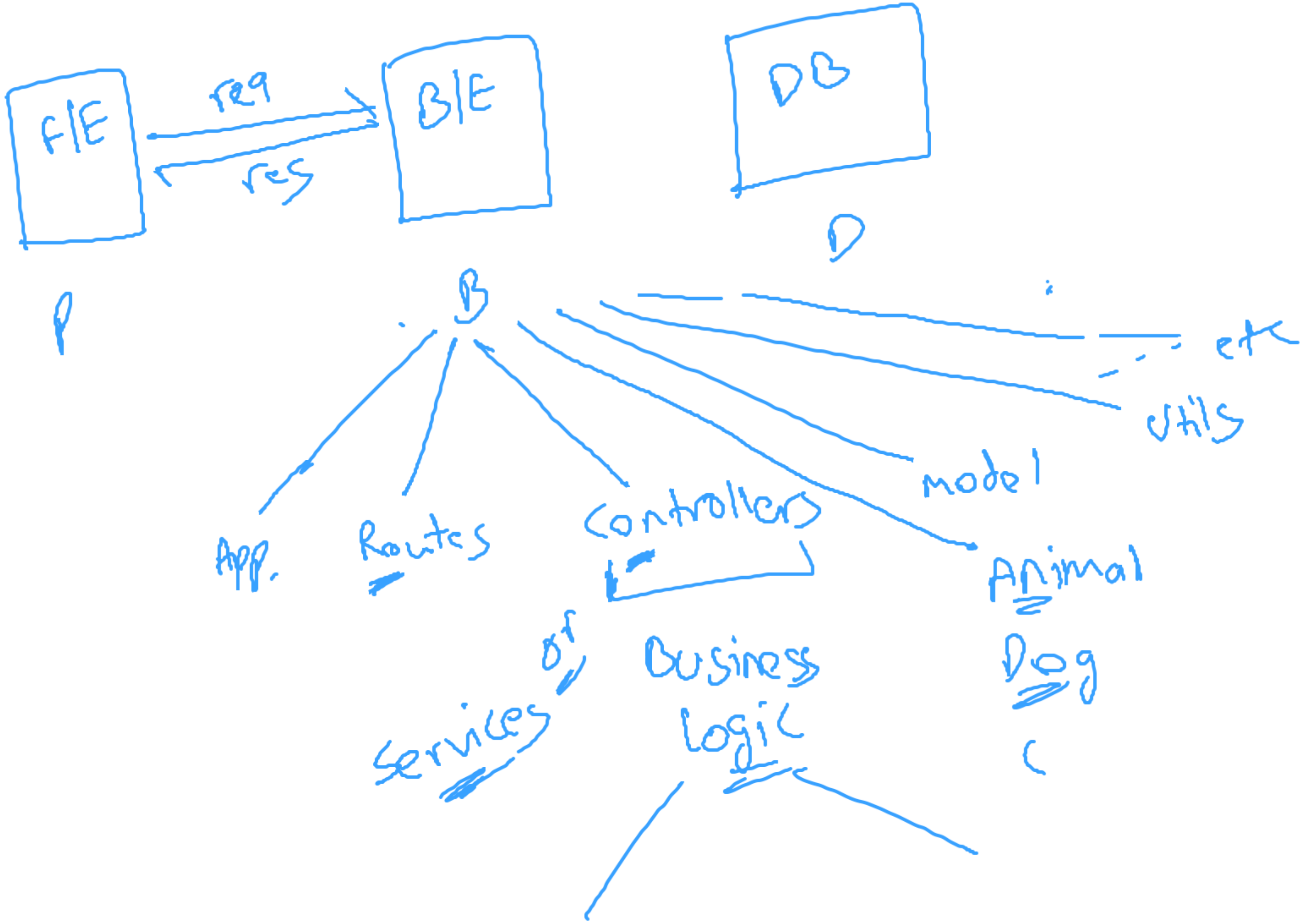
Example

IAnimal (15 functions)

- Dog
- Animal
- 100 more

Method Overloading
- Same Name (Function), but different signature(params, return)

LLD
(code level)


F/E
req
res
B/E
DB
F
B
D
App.
Routes
Controllers
or
Services
Business
logic
model
Animal
Dog
C
utils
etc