OOPS:

Study in class →

1. classes and objects
2. Data member
3. member function
4. Access Modifier
5. constructor

6. Static keyword
7. string [ ] args
8. this keyword
9. classes [Default package]. linear list
   - stack
10. Interface

game with OOPs

Learn. (for interview)

1. Polymorphism → Runtime
   → compile time
2. Encapsulation
3. Abstraction
4. Multi threading → Synchronization
5. Inheritance → + interface
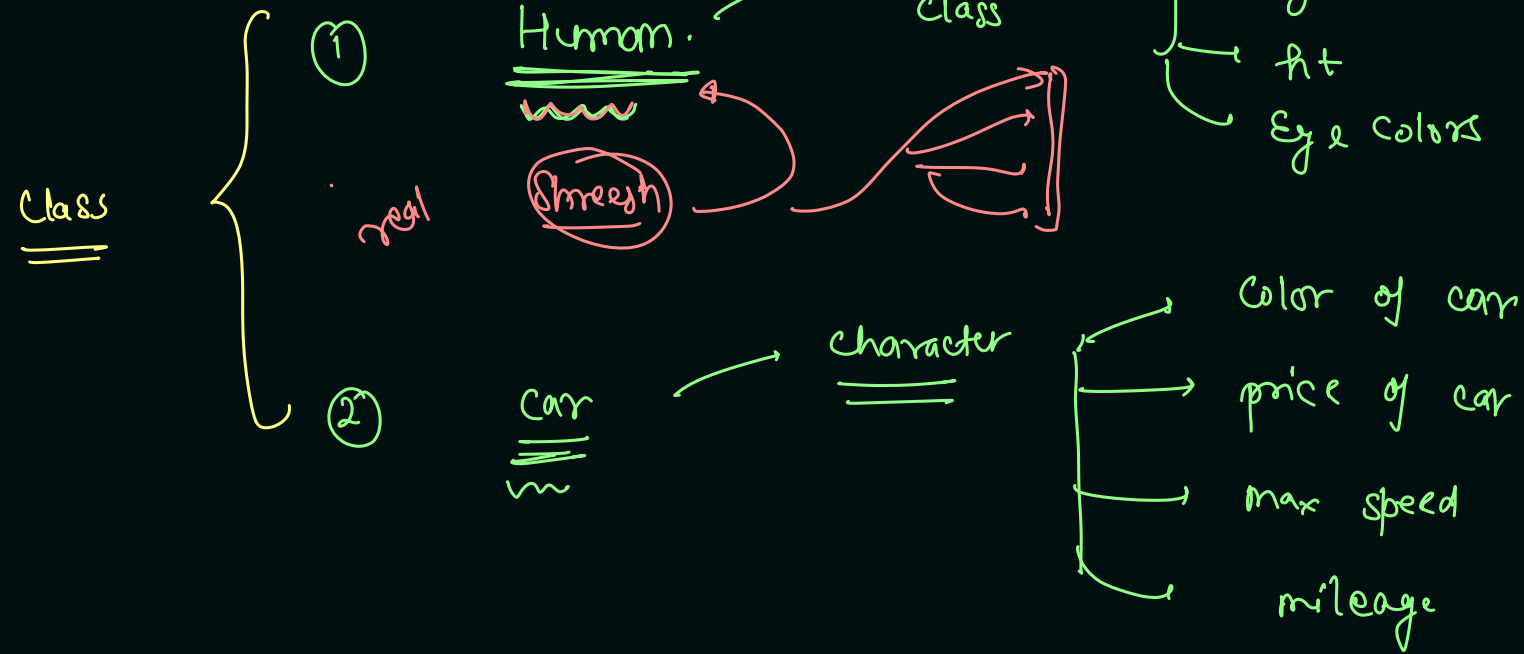
6. super
7. constructor. , what destructor? → C++
8. this
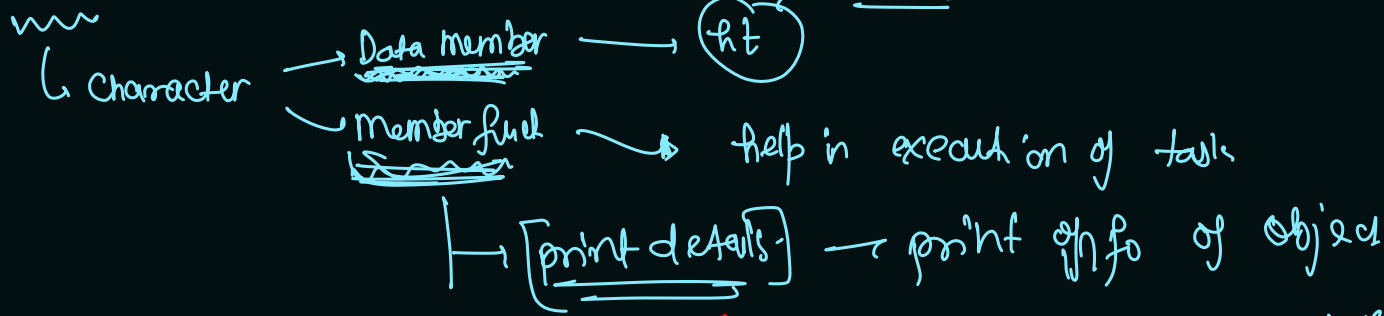9. Exception handling.
10. interface

Classes and objects:

└─→ Blue print of real world entity.

Class
{
① Human ──→ Character in class ──┬→ skin color
│   ├→ Age
│   real   (Shreesh)   ├→ ht
│   └→ Eye colors
│
② car ──→ character ──┬→ Color of car
                       ├→ price of car
                       ├→ max speed
                       └→ mileage
}

Object → A real world entity that belongs to a particular class.

Data Member, Member function:

∿∿

└ Character

→ Data member ──→ (ht) → double] contains decimal

→ Member func ──→ help in execution of task

└─[print details] → print info of object

Java → [.java] compile
Code → class] ── (.class)
C++] x ← ↓ run
using
package. A   JAVA

→ File A
   File B
   File C

com be
a
class
com
be
interface

Access Modifier :

Reach of data members
member function
└ Accessibility.

nested class
can be private
data member,
member func
Package B

C lau →
fn A
fun
Clau B func
Subclass
Inheritance

Access	default	private	protected	public
Same Class	Yes	Yes	Yes	Yes
Same package subclass	Yes	No	Yes	Yes
Same package non-subclass	Yes	No	Yes	Yes
Different package subclass	No	No	Yes	Yes
Different package non-subclass	No	No	No	Yes

Static

Static int (a) → Stack Segment

→ address → Static Segment

fun( ) {
  a = 20;
  Syso(a);
}
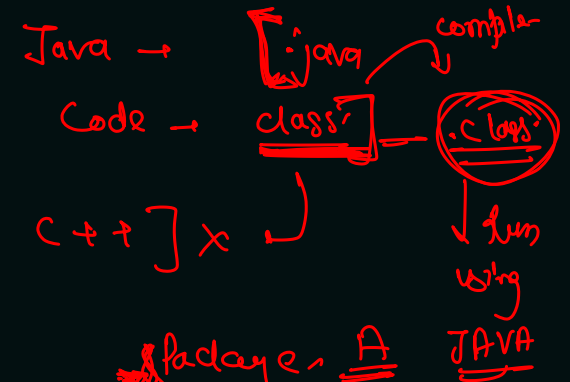
may be or
may not be
static

Stack Memory

Static Segment
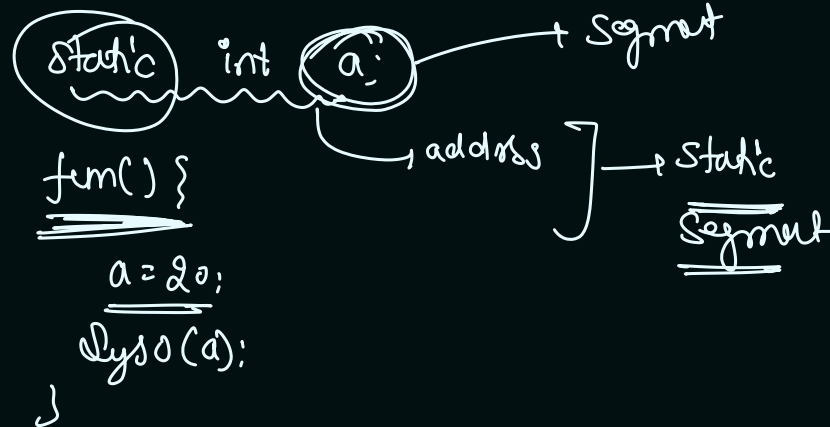
fun 2 ( ) {
  a = 30;
  Syso(a);
}

Static → class
  ↳ method
  ↳ variable

(Method) — Body of method required some kind of memory.

Static —— Non accessible without object —→ Non Static

accessible or Static
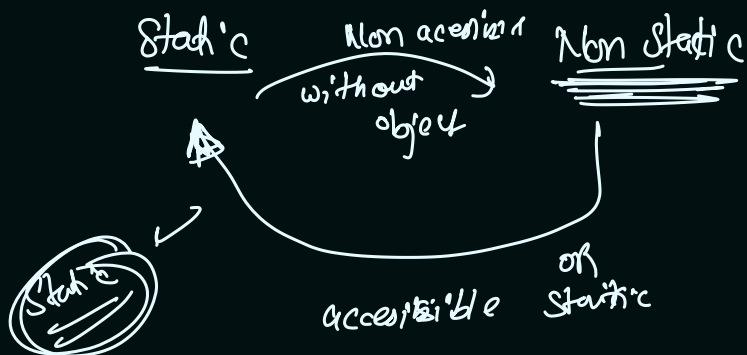
Constructor : 1 default] = No-args in C++
            2 parameterised] = Con.t
                                    { destructor] → Not in JAVA
            3 copy constructor]
                                    { Rather than destructor, JAVA use
                                      garbage collectn -

if Explicitly no constructor is   Garbage collectn → Random fine of garbage
written ,then JVM provide   collector
                                    |→ Mark &
default constructn to   Swipe Algorithm,
the class,   Refern
                                varia   Memenn
                                v1 _____ m1
                                v2 _____ m2
                                v3 _____ m3
                                        [Remainly Memory]____ delete-
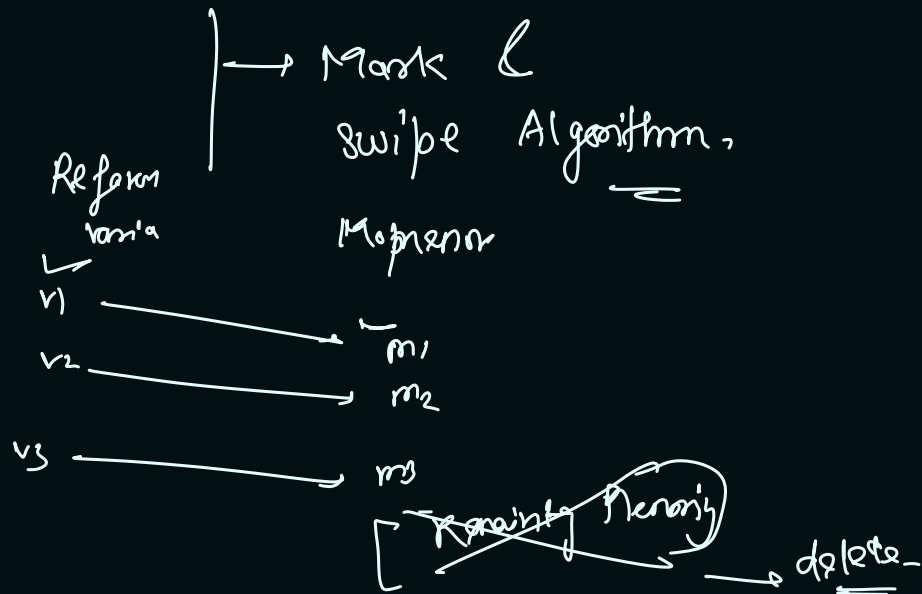
    More than one constructn can be written in class-

this keyword:

this → self Referencing pointer.

this]→ always pass with <u>object call</u>
                              └→ Member function

```
void fun () {
    ⟨this⟩━━
    
    ⟩
```

d1. fun();

user defined class ──→ <u>Class C1</u>]→ {← extend object class
                                              └ By default
                                                 done by JVM
├→ notify ()
├→ hascode()
├→ sys°(a) ──────── +destroy ()
├→ equals()

Interface :- Contract] → Sign class

4 yrs. → 1 machine

Classes → Extend
Interface → Implements   learning - L1

L2 → Question
         Bank

All topic in DS

Go funct'

Interface   comparable <E>

P. Int   compareTo(Eo) {
          ────── No function Body.
}
}

tries
Bits
AVL
Backtracking

Prepenation

Complete.

Company
Specific
Revision
during
Company test
or
Interview

Level 1 & Level 2

Interview Experiance

Class   Car   Implement   Comparable < Car > {

Class body

compare To ( ) {

}
}

Amazon → LL & tres
Microsft ──→ Graph
                      &
                      DP
Google → Graph & DP
                Bits. & trie