↳ Object Oriented Programming

Types of prog. Language

Procedural   OOPS
    C   C++, Java, Python

Abstraction Encepsubtion
            ┌─────────┐
      (──→ │ OOPS   │ ──→ Inheritance
            └─────────┘
                      ↳ Polymorphism
   class   object
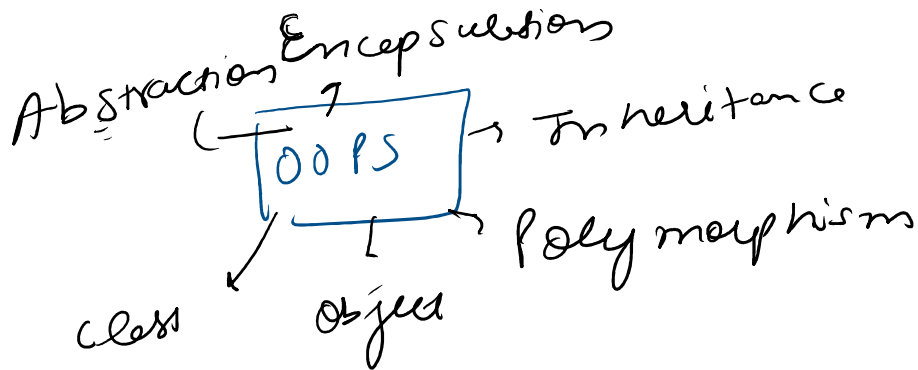
Pillars of oops

Class ↳ blueprint of a real world entity called object -

Object ↳ Real world entity | physical entity
                    ↳ it gets memory

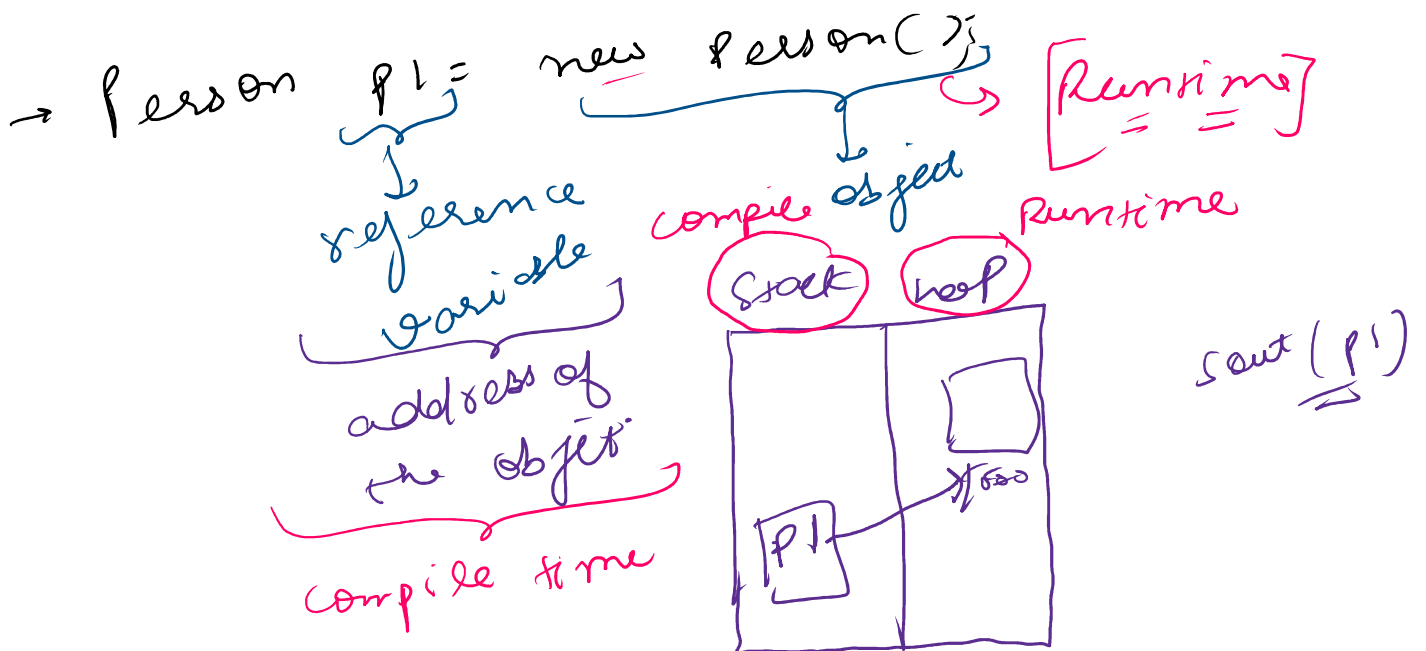class Customer {
  string name;
      email;
}

```
String name;
String email;
sayHello() {
    sout("Hello");
}
}
```

Customer Kunal = new Customer();
Customer Kohli = new Customer();

→ We can create as many objects as we want for a class.

→ Person P1 = new Person();

reference variable

address of the object

compile time

compile object

Stack   Heap

Runtime

[Runtime]

Runtime

sout(P1)

```
class Person {
    String name;    } → member
    int age;        }    variable

    func() {   } → member
    () {       }    methods
}
```

func() }
   (
3   }   } → men...
                              methods

→ Class name always starts with capital letters.

Constructor
Is first method which is called by your compiler (JVM) to create the object.

→ Java adds a default constructor to every class;

→ Constructor's name is always same as the class name

Person ( )

→ Constructor doesn't have a return type

Person(){
   (,
   }

Ⓐ Its either the programmer or Java who will

(A) Its either the programmer or ... . write the constructor. Dono ek saath nhi likh enge.

→ Constructor overloading
    ↳ multiple constructors with diff parameters.

this keyword
    ↳ 'this' refers to the current object.
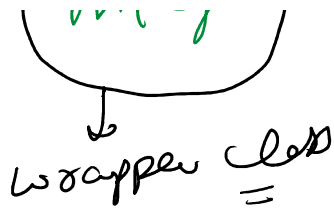
Q) Is Java a 100% OOP language?
    If yes/No → why?
    → No, Because of primitive data types.

$$[int\ a = 10];$$

Integer a = new Integer(1);
    int a = 10;

wrapper class

int a = 10;
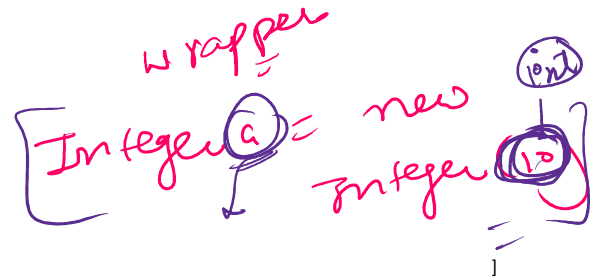
int → Integer
float → Float
double → Double
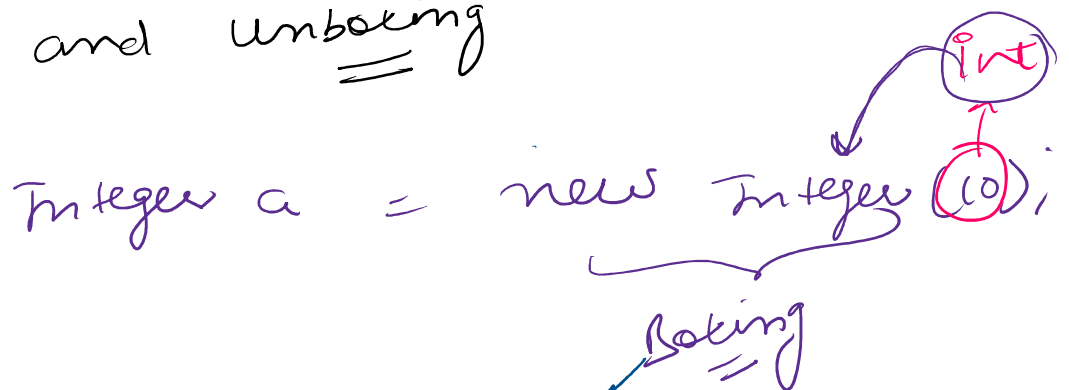char → Character
boolean → Boolean
short → Short
byte → Byte
long → Long

wrapper

Integer a = new Integer(10);

Ⓟ There is no concept of pointers in Java

Boxing and unboxing

Integer a = new Integer(10);

Boxing

Boxing

converting a primitive into object.

int { b = (a) → obj } unboxing
                          ↓
                       converting object
                       into primitive.

[→ Auto Boxing / Auto unboxing
]   ↓
         v.5 of Java]

Constructor chaining
              ↓
    calling one constructor from another
                  constructor

⊛ Constructor is called only once for an object in its entire _life_ time.

Copy constructor
↳ a constructor which takes self object in the input and creates a new object

Static keyword
→ variables ⎱
→ methods ⎰ ═══
→ class ⎤ later

⤷ Static is used to share things across all the objects
  ↳ doesn't belong to any object
  ↳ belongs to a class...

→ static methods and variables belong to the class and not the objects.

(P) Static members can be accessed inside a non-static method.

→ Java by default follows CamelCase

[firstName]   (age)   ageValue

→ SQL → snake_case → first_name