

Function ✓

C → Procedural Language (PPL)
↳ function through
saara kaam

$f_{100}(n) \leq$

$$fib_0(n-1) + fib_0(n-2)$$

Why were C++ or Java /OOPL?

Nouns (Objects)
Person → Person

Object → Noun

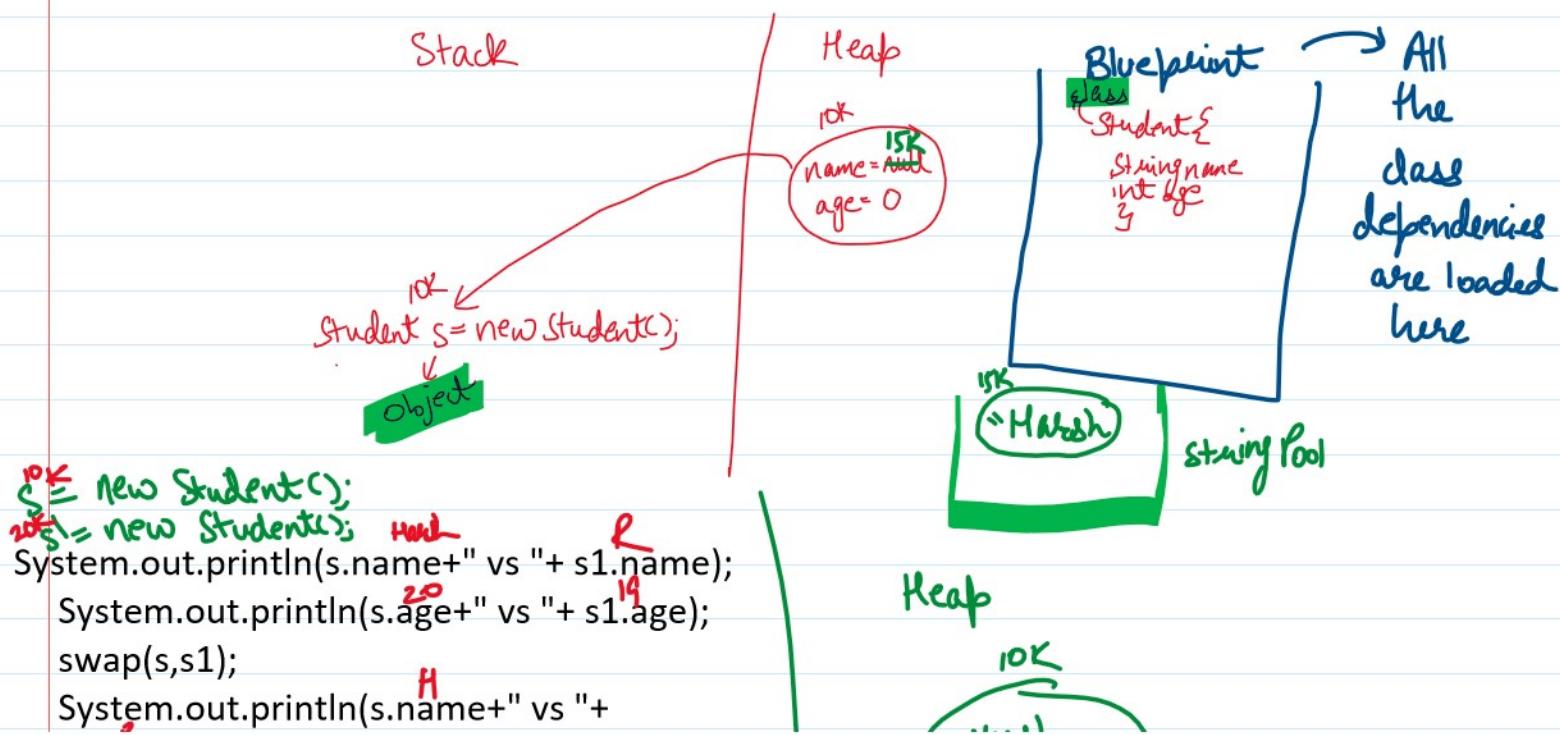
- 1) Name
- 2) Age

Person

- ↳ Marsh, 20
- ↳ Miller, 21
- ↳ Ragnar, 22

class Student {
 String name;
 int age;
}

Student s = new Student();
Object +
 Non Primitive
 ↳ Heap



```

swap(s,s1);
System.out.println(s.name+" vs "+  

s1.name);
System.out.println(s.age+" vs "+s1.age);
}

```

```

public static void swap(Student s1, Student  

s2) {
    Student temp = s1;
    s1 = s2;
    s2=temp;
}

```

Swap2

$s = 10K$
 $s1 = 20K \quad H20$
 $R19$
 $\text{sysout}(s.name, s1.name);$
 $\text{swap2}(s, s1);$
 $(\text{sysout}(s.name, s1.name);$

```

public static void swap2(Student s1, Student s2) {
    String temp = s1.name; ✓
    s1.name = s2.name; ✓
    s2.name=temp;

    int temp = s1.age;
    s1.age = s2.age;
    s2.age=temp;
}

```



Jiske through • Rete ho , uska address context

mein pass ho jata hai



→ Keyword

↳ Address access kar skte ho kisi bhi

object ka

initializing the variables

- 1) Data parsing → data members ki koi default value gets picked
- 2) Constructor → A special function with no return type having same name as the class.
↓
return; ?

return type is class ka object

Student s = new Student()

← ↗
returns constructor
address of
object

Steps

- 1) new Student()
- 2) Data parsing
- 3) constructor

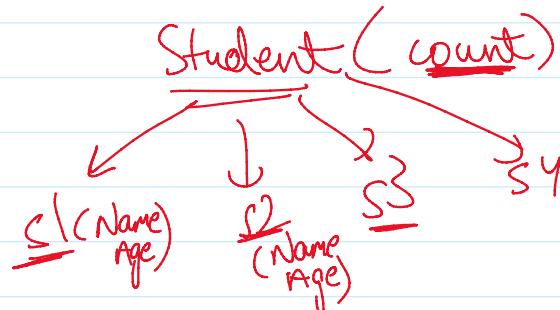
Kitne students hain?
↳ objects

Object

(50k)
stud = [50k
= [10k, —, —, —, —, —, —, —, —, —]
new Student() ✓

static → Object ki property ✗
 variables
methods
blocks

class ki property ✓
 Static
 class bind sc



Access Modifiers

1) Public → can be accessed from anywhere

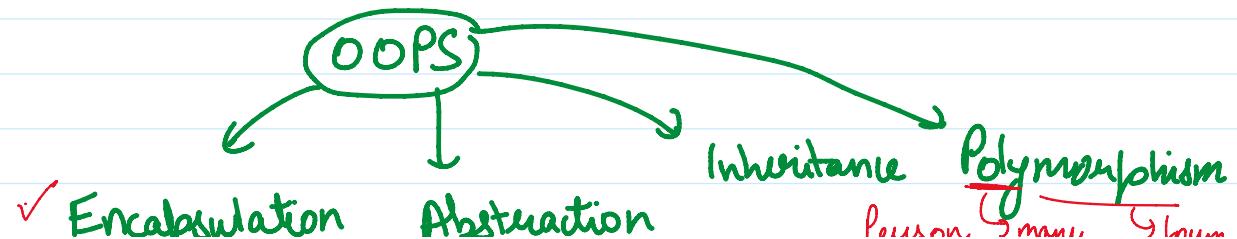
same class ✓, same file
 same package ✓
 same project ✓

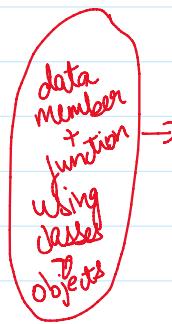
2) Private ⇒ can be accessed from same class only

3) Protected ⇒ can be accessed from same class
 & inherited class

4) default → can be accessed from same class ✓

same file ✓
 same package ✓
 same project ✗





Wrapping up of data members & function under a single unit known as a class

+ Access Specifiers / Modifiers
 ↗ (Public, Protected, default, Private)

* Advantages of OOPS

- Better model for real life applications
- Provides more security
- Reduces redundancy
- Modularity ↑



1) Kal
 ↓ Doubt ✓
 10

9

Syllabus
 ↓
 9 → 1 ✓ 20 ✓

Weekday
1 online

O✓
 N✓
 D✓
 S✓
 March
 End

20

4
 8