

OOPS-2

constructors
shallow vs deep
pass by value
static keyword.

Student S = new Student ();

Constructor

↓
helps us to initialize
the fields.

class Student {
 age
 gradYear
 public Student () {
 //
 }

parameters ?

}

Student (int age, int gradYear) {
 age = age;
 gradYear = gradYear;
}

}

req: I will give you a Student object,
you need to create another object
by copying the data.

```
Student st = new Student();
```

```
st.age = 25;
```

```
st.gradYear = 2020;
```

```
Student s2 = s1;
```

↓
ref copy

X new object
is not being
created.

```
Student s2 = new Student();
```

```
s2.age = s1.age;
```

```
s2.gradYear = s1.gradYear;
```

what if
I need copying in
12 diff files?

```
Student (Student other) {
```

```
this.age = other.age;
```

```
this.gradYear = other.gradYear;
```

```
Student s2 = new Student(s1);
```

```

Student {
    int age;
    int gradY;
    Exam enrollExam;
}

```

```

Student ( Student other) {
    this.age = other.age;
    this.gradY = other.gradY;
    this.enrollExam = other.enrollExam;
}
}

```

Diagram illustrating the shallow copy in the constructor: A red arrow labeled "copied ref" points from the `other.enrollExam` field to the `this.enrollExam` field, indicating that the reference is copied rather than the object being cloned.

```

Student st1 = new Student();

```

```

st1.age = 25;

```

```

st1.gradY = 2020;

```

```

Exam exam = new Exam();

```

```

exam.id = 1;

```

```

exam.score = 100;

```

```

st1.enrollExam = exam;

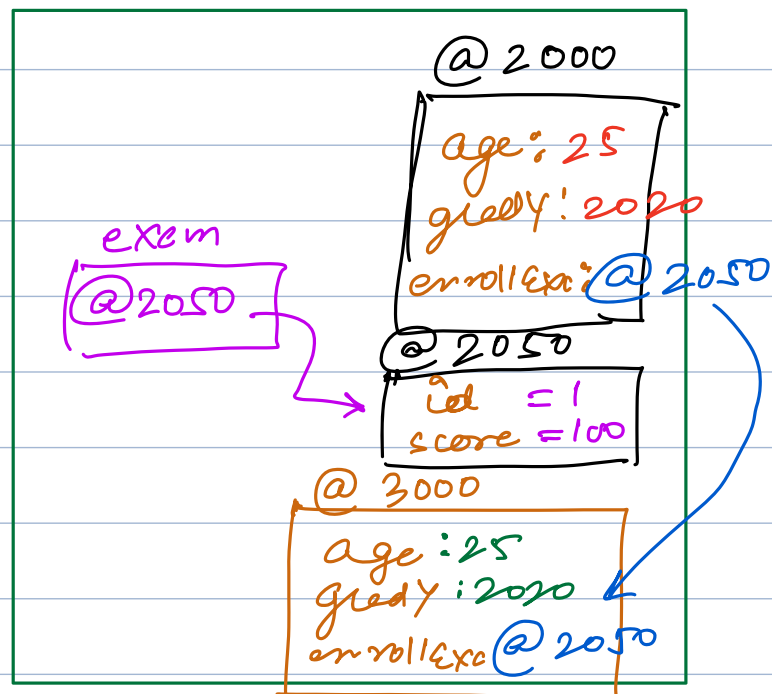
```

```

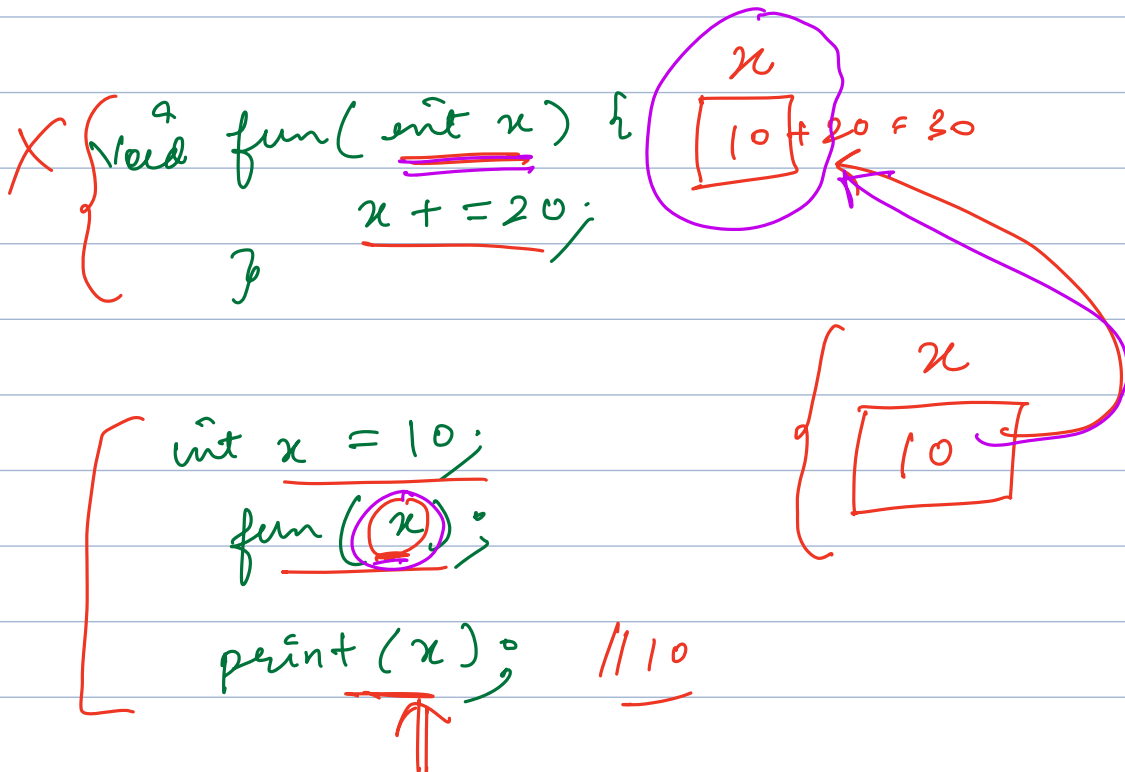
Student st2 = new Student
                (st1);

```

SHALLOW



Pass by value, pass by reference.



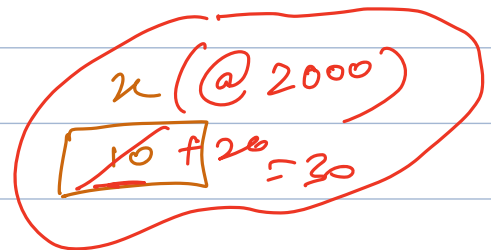
C++

void fun(int &x) {
 x += 20;
}

pass by reference

int &x

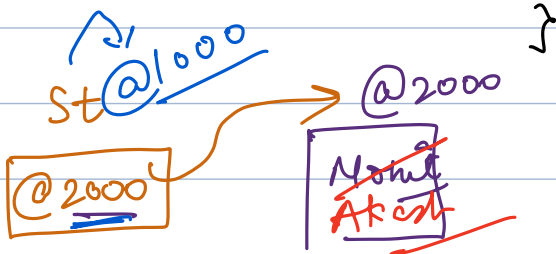
int x = 10;
fun(x);
print(x); // 30



```
fun( Student st ) {
```

st
@2000

```
    st.name = "Akash";
```



```
Student st = new Student();
```

```
st.name = "Mohit";
```

```
fun(st);
```

```
print(st.name); // Akash
```

```
fun(@2000);
```

Static



```
class Student {
```

```
    int age;
```

```
    static int x;
```

↓

common to all
of the objects.

all of the objects
of the same class
to use one
variable.



class variables

→ constructors

- default
- parameters

- copy
↓
deep vs shallow
value vs ref

→ static

↳ variables
methods

HW1: static blocks?
HW2: When static variables get initialized? — class loading

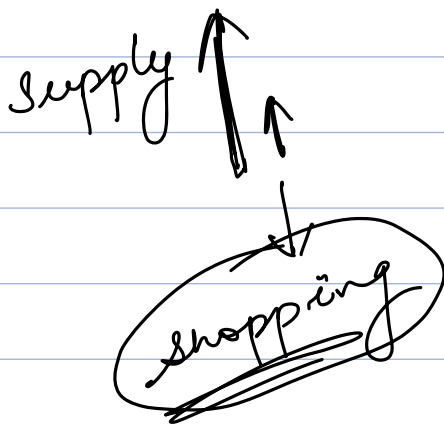
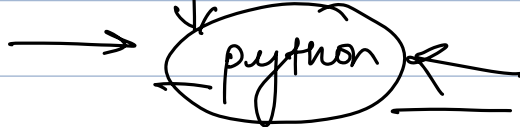
Inheritance

Fridays / Polymorphism

Doubts : OOP-I] Assignment]

Homework \rightarrow PSP X

Java



Ass + HW

BS+I

[google, chatgpt, claude
had, google]