

## Agenda:

① Recap

② Constructors

③ Access Modifiers

primitive data types → int  
boolean  
float  
double  
char

classes help us create our own custom data

types

class Student {  
 int age;  
 double psp;  
} → define data types  
→ blueprint

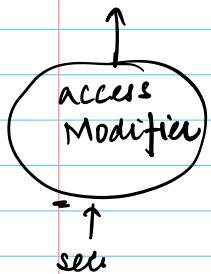
int  
double  
float }  
int a = 5; } → declaration ]  
object ↑ → initialization ]  
↓ instance of int  
int a; ↓  
Student a; ↓

int  
Integer  
JVM  
method area

a = 10  
a = new Student()  
declaration

```
package X;
```

```
public class Student {
```



```
int age;
```

```
String name;
```

```
int problemsSolved;
```

} → declaration

```
void solveProblems ( int n ) {
```

```
problemsSolved += n;
```

```
print ( problemsSolved );
```

```
}
```

main fu

```
student A = new Student();
```

```
A. solveProblems( 1 );
```

initialization

→ 1<sup>st</sup> Line: declaration, initialization

=  
output = 1

new → creates space  
in memory  
for object

String → not a  
primitive  
type

Non-primitive type objects

A

student	
age	0
name	null
problemsSolved	0
solveProblems	0

DEFAULT

have default  
value = null

## CONSTRUCTORS

Special method that is called when  
an object is instantiated.

- ① Name of constructor == Name of class
- ② invoked automatically at the time of object creation
- ③ No explicit return type. → Why?  
↳ Return an object of Type  
Student



How were we able to create objects in  
the previous example?



default constructor

[ Student( ) { } ]<sup>o</sup>

## Default constructors

↳ → implicitly defined for each class

→ returns a new object of that class with  
value of all attributes as the default  
value of that type

If you prefer to initialize with ~~different~~<sup>fixed</sup> values?

### Approach - 1

Write down the values  
in a custom  
constructor

### Approach - 2

Initialize variables  
while defining

Would the default constructor still be available?

utils  
↳ static  
method

```
private Student () {  
    name = "Meh"  
    age = 100  
    problemsSolved = 2000  
}
```

Now,  
default  
constructor  
is no  
longer  
available

↑ [what if I want to create a class that  
cannot be initialized.]

## PARAMETRIZED CONSTRUCTOR

Till now, we initialized objects with default or fixed values.

Now, I want to initialize objects with custom values

⇒ Pass value at the time of constructing objects

```
Student a = new Student( )  
student a =  
new Student("a", 2);  
public class Student {
```

int age;

string name;

int problemsSolved;

void solveProblems( int n ) {

problemsSolved += n;

print( problemsSolved );

student( string( ), int( ) );

name = x;

age = y;

student( string( ),  
int age ) ;

name = name  
age = age

"This" keyword

3

Object's  
Attributes

[ name - space collision ]

this

Student ( string name, int age )

this.name = name;

this.age = age

student a = new Student ("a", 10);

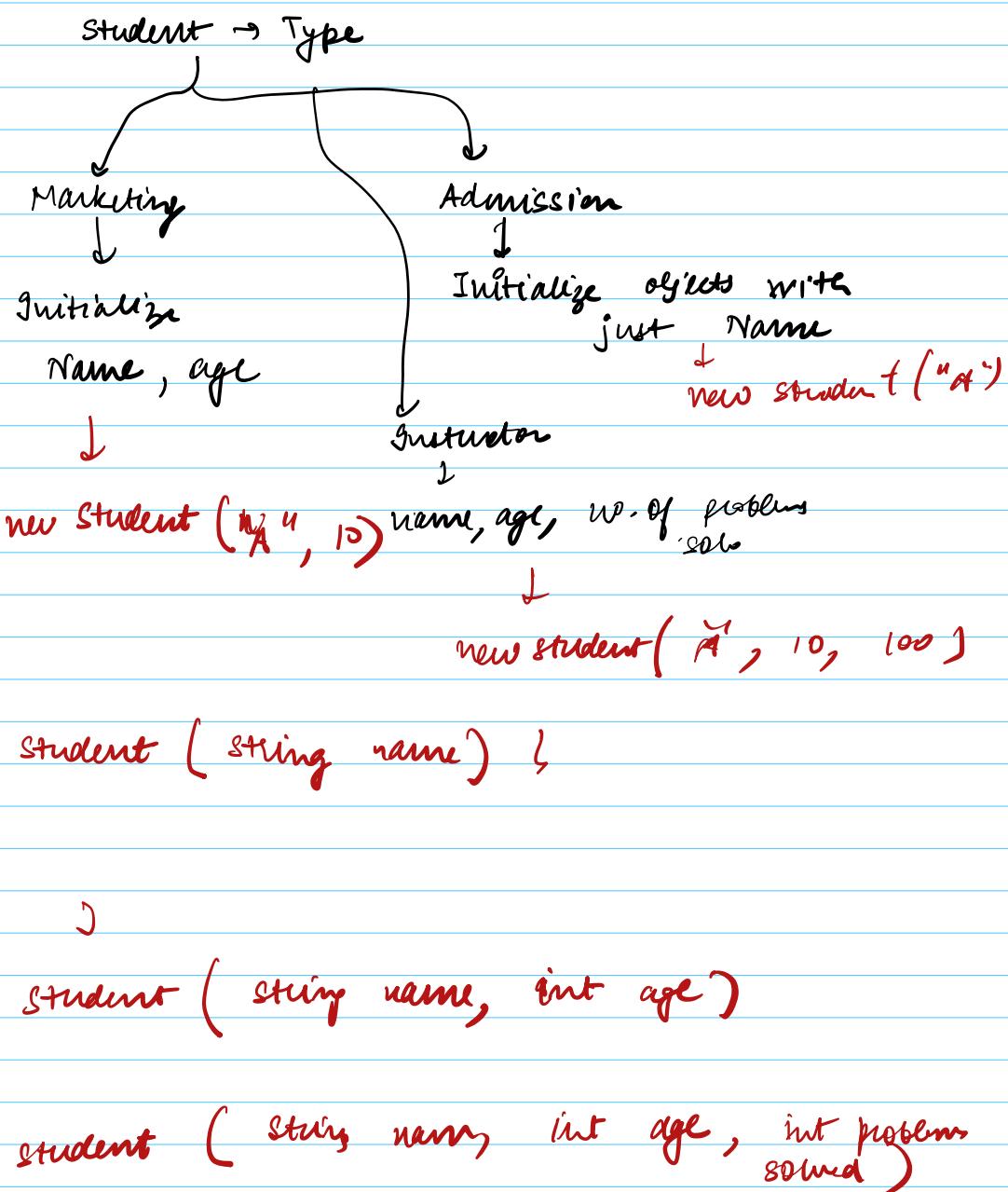
To invoke method  
or attribute  
of the object  
for which  
the constructor  
is called.

"this"

name  
age  
problem  
solveProblem()  
student a 10,

a.name

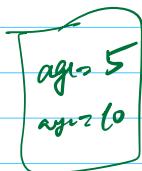
## Overloading Constructors



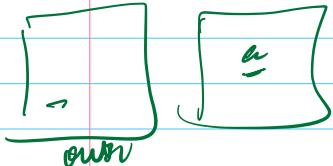
→ Break - 5 mins

copy constructor

10:15



public void modify ( int age )  
age = age + 5;



int age = 10

modify (age)

print (age) → 10 NOT MODIFIED

@120  
public void modify ( student s ) {

s.name = "Rahul";

student s  
modify (s)  
print (s.name) → Rahul  
↳ modified

a → 5

int a = 5

(a) → 5

Reference

Address of

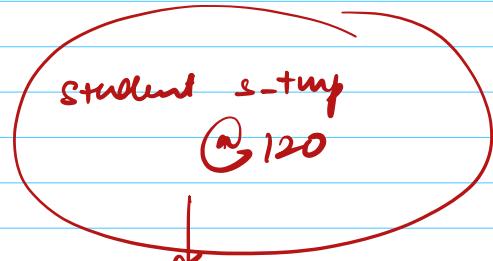
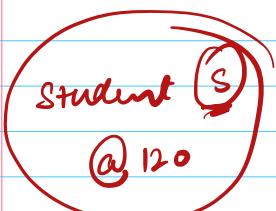
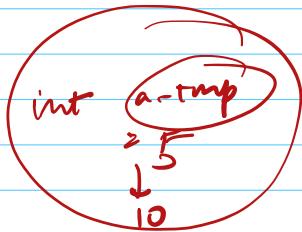
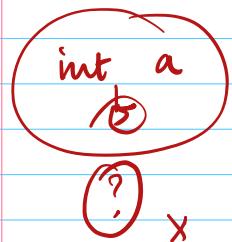


Always passing copy of value to methods.

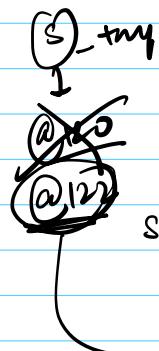
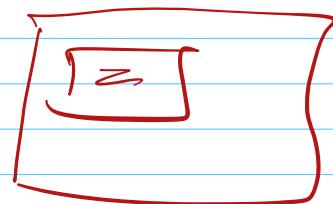
Parameter  $\neq$  Value itself.

new place  $\equiv$

Reference



s-tmp. name = "Kahn"

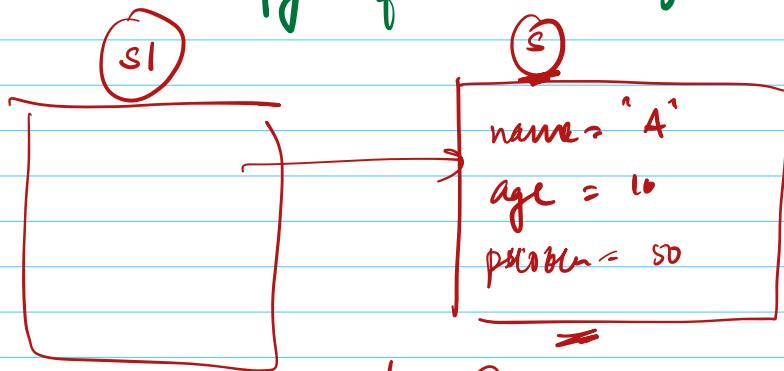


s-tmp = "new-name"

copy constructor

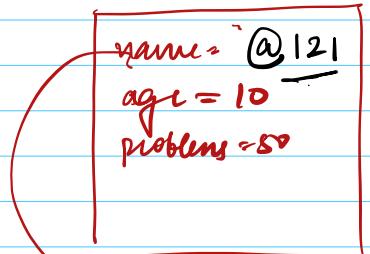


construct an object as the  
copy of another object

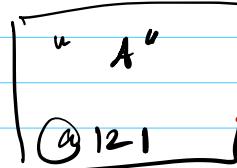
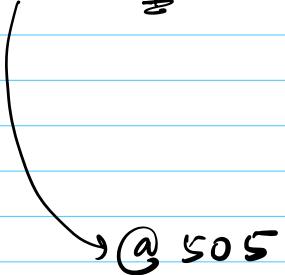


student ( student other )

student s1 = new student("A", 10, 50)



student s2 = new student(s1)

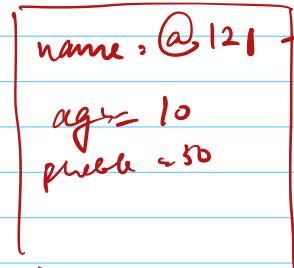


student( student other ) {

this.name = other.name

this.age = other.age

this.problems = other.problems



}

shallow