

Agenda

Class

Objects

this → Keyword

Memory

Constructors

What not to discuss
Static

OOPS

Access modifiers

↳ public

↳ private

↳ protected

Data types

int] Integer

long

float

double] Deci

char

→ Single character

boolean

→ true/false

String

↳ seq. of characters

Store data of an entity →

car

- ↳ color → String
- ↳ model → String
- ↳ year → int
- ↳ speed → int
- ↳ price → int

Student

- ↳ name → String
- ↳ attendance → int
- ↳ DOB → String
- ↳ gender → String
- ↳ address → String
- ↳ phone no. → long
- ↳ psp → double

user defined
datatype

what is class ?

Blueprint of a real world entity

Student

↳ Name

↳ DOB

↳ PSP

↳ phone No.

Student



class
user defined
datatype

s1 ;



object /
Instance of
that entity

Syntax →

static class name-of-class {

// 1. data member

// 2. member function

// 3. constructor

}

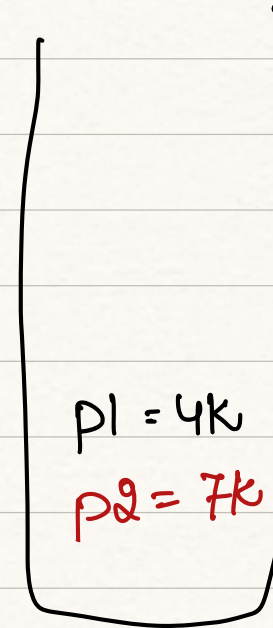

```
static class Person {
```

```
    String name;  
    int age;
```

```
}
```

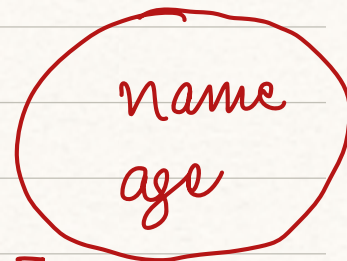
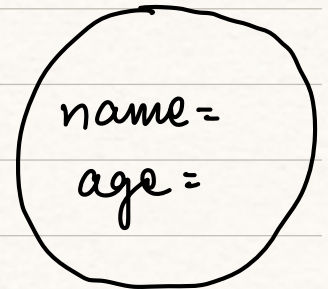
```
Person p1 = new Person();
```

```
Person p2 = new Person();
```



Heap

4K



7K

blueprint

```
class Home {
```

```
    String address;
```

```
    int roomCount;
```

```
    boolean furnished;
```

```
    boolean parking;
```

```
    int balconyCount;
```

```
}
```

```
Home A = new Home();
```

```
A.address = "xyz";
```

```
A.roomCount = 4;
```

```
A.furnished = true;
```

```
A.parking = true;
```

```
A.balconyCount = 2;
```

```
Home B = new Home();
```

```
B.address = "abc";
```

```
B.roomCount = 3;
```

```
B.furnished = false;
```

```
B.parking = true;
```

```
B.balconyCount = 1;
```

```
class Home {
```

```
String address;
```

```
int roomCount;
```

```
boolean furnished;
```

```
boolean parking;
```

```
int balconyCount;
```

```
}
```

```
main () {
```

```
Home h1 = new Home();
```

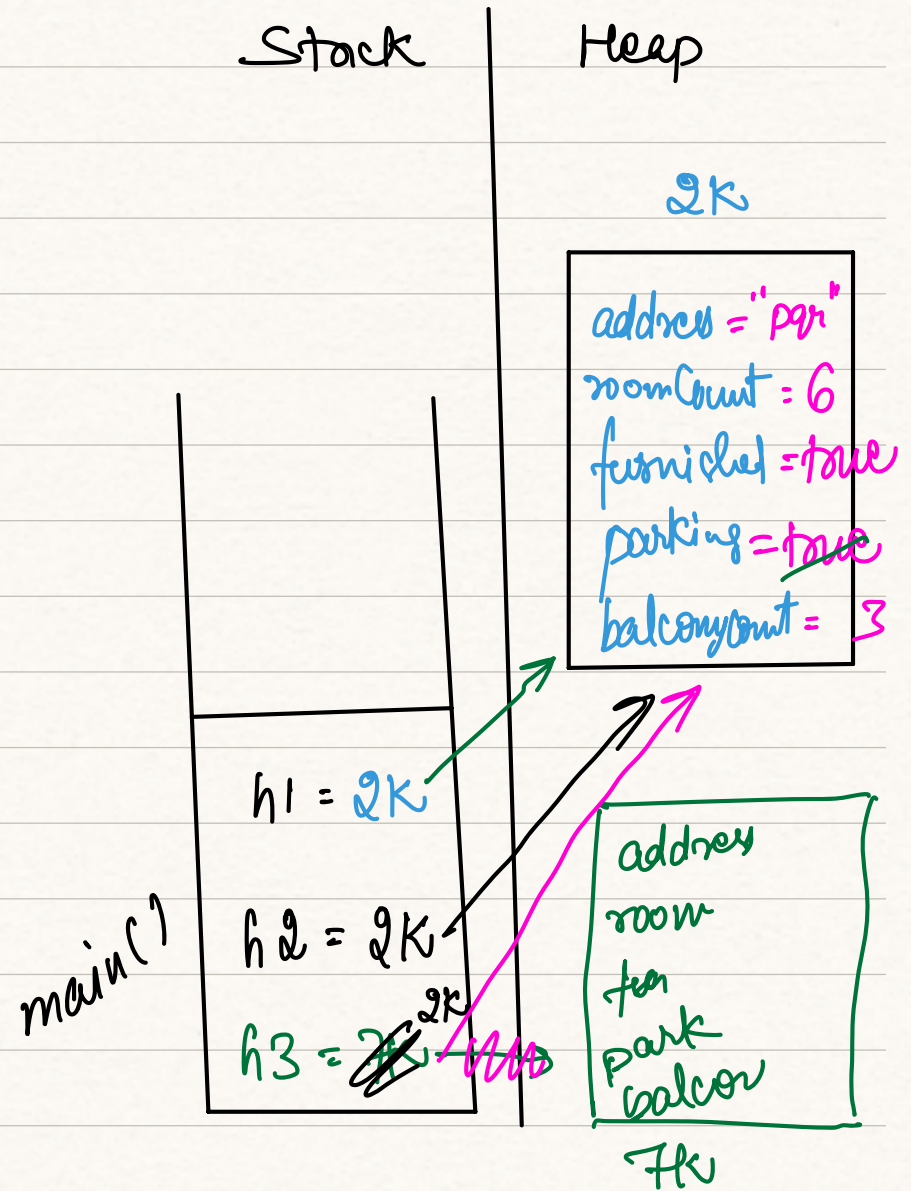
```
h1.address = "pqr";
```

```
h1.furnished = true;
```

```
h1.roomCount = 6;
```

```
h1.parking = true;
```

```
h1.balconyCount = 3;
```



Home h2 = h1;

h2.parking = false;

] not the correct
way to copy

Home h3 = new Home();

h3 = h1;

] not the
correct way

}

What is constructor?

A function without a return type
The name of this function is same as class.

static class Employee {

String name;
int age;

① Employee (String x, int y) {
}

Zero argument
constructor
{

← Employee () {
② }

name = null
age = 0


```
Employee e1 = new Employee ();
```

```
e1.name = 'Kapil';  
e1.age = 25;
```

e1.name = null

```
Employee e1 = new Employee ("Kapil", 25);
```

The default constructor will be called when you do not have constructors in class.

class → blueprint of a real life entity

Data members
member function
constructors

you can create multiple instances of a class → objects

```
class Car {  
    String color;  
    int year;
```

```
Car c1 = new Car();
```

default constructor

```
Car c2 = new Car();
```

"this" → self reference

```
boolean isValid (int curr) {
```

```
    if (year + 15 > curr) {  
        return true;  
    }
```

```
    return false;
```

```
c1.color = "white";  
c1.year = 2021;
```

15 year

}

}

System.out.println (c1);

S.O.P. (c1.isValid (2023));

↓
address of instance/
object

type of constructors →

Default constructor →

If you do not write constructor you get
to use default constructor.

Note → If you have written constructor in class
then default constructor does not
exist.

Zero argument constructor → having no arguments

Parameterized constructor →

having some arguments;

```
class Car {  
    String color;  
    int year;
```

zero
argument

```
Car () {
```

```
}
```

```
Car (String color) {
```

```
}
```

```
Car (String color, int year) {
```

```
Car c1 = new Car ();
```

3

2