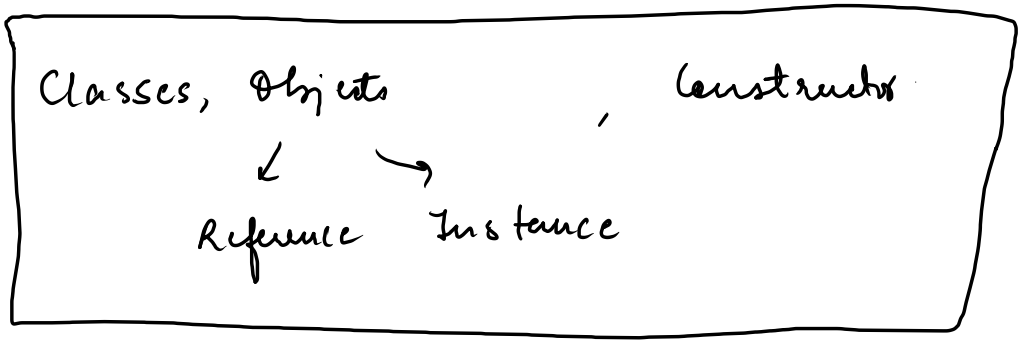
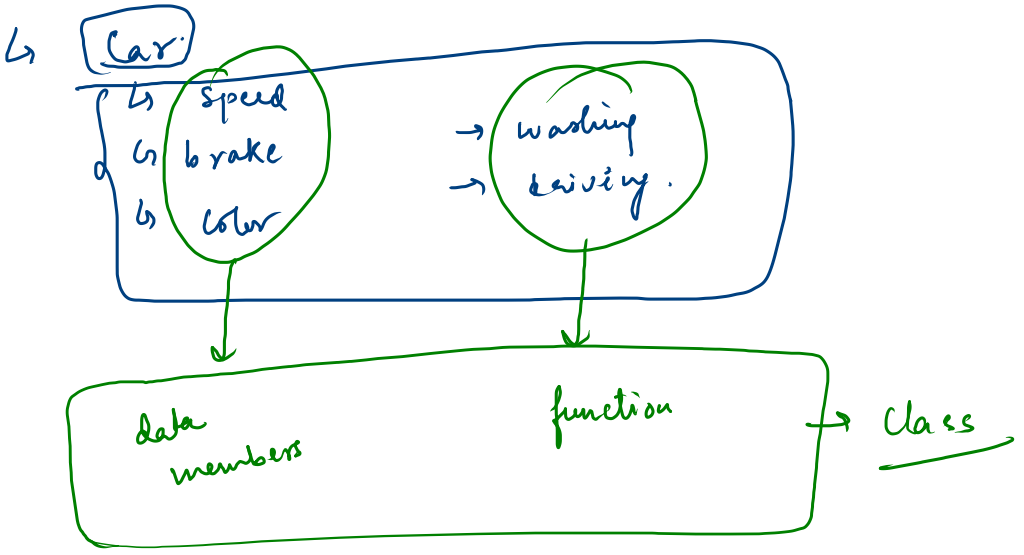
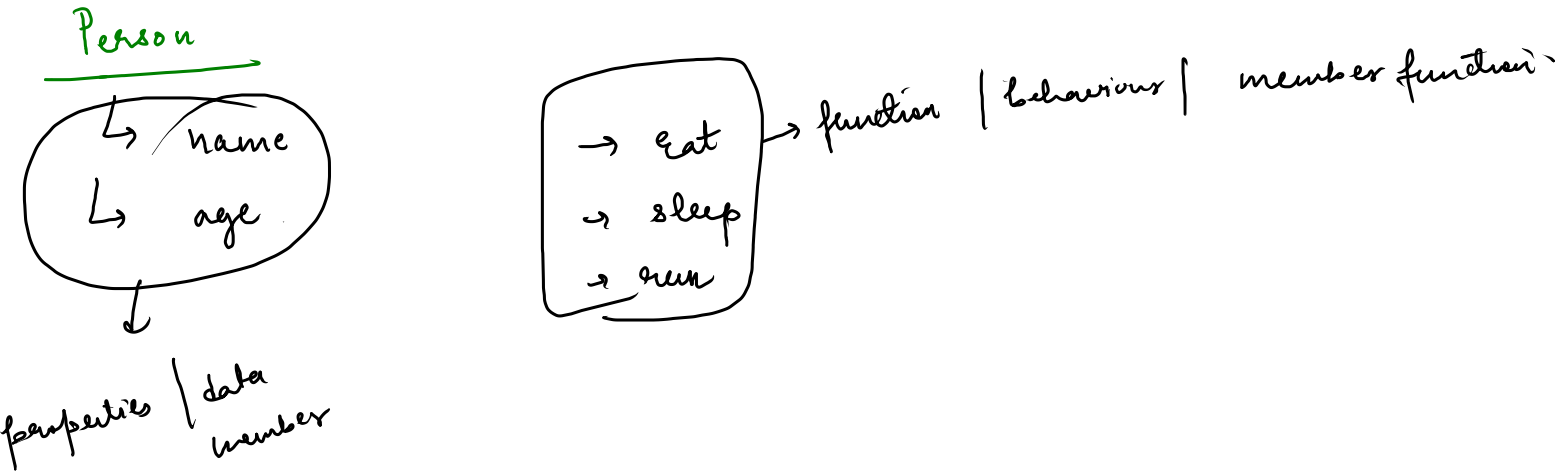

Classes, Objects
Constructor
Reference
Instance

* Class :-

↳ blue print of creating object.

↳ putting data members and related fns. together.




Person
name
age
properties | data member
eat
sleep
rem
function | behaviours | member function

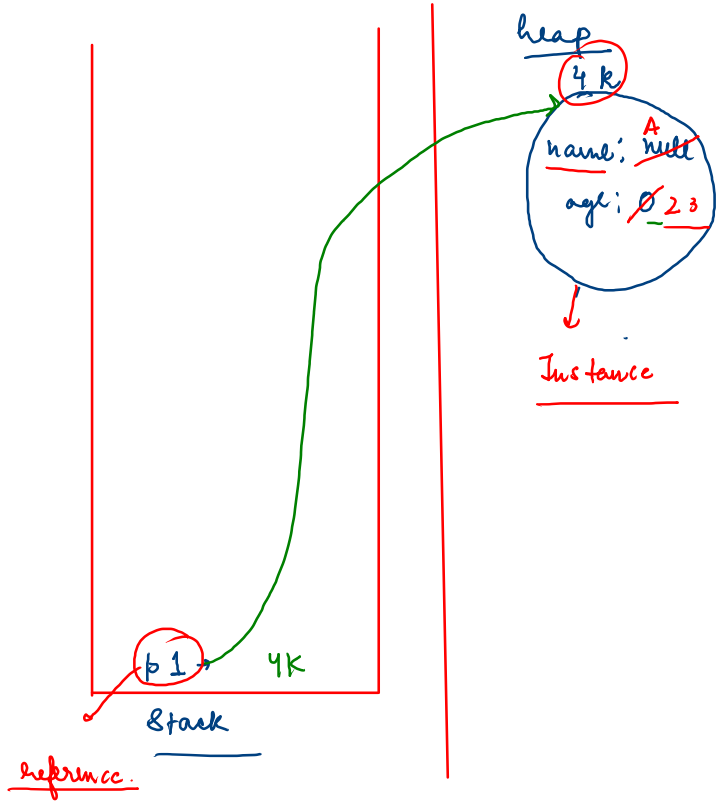
```java
public static class Person {
    String name;
    int age;

    public void saysHi(){
        System.out.println(name + "@" + age);
    }
}

public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN.
    Person p1 = new Person();
    p1.name = "A";
    p1.age = 23;

    p1.saysHi();
}
```

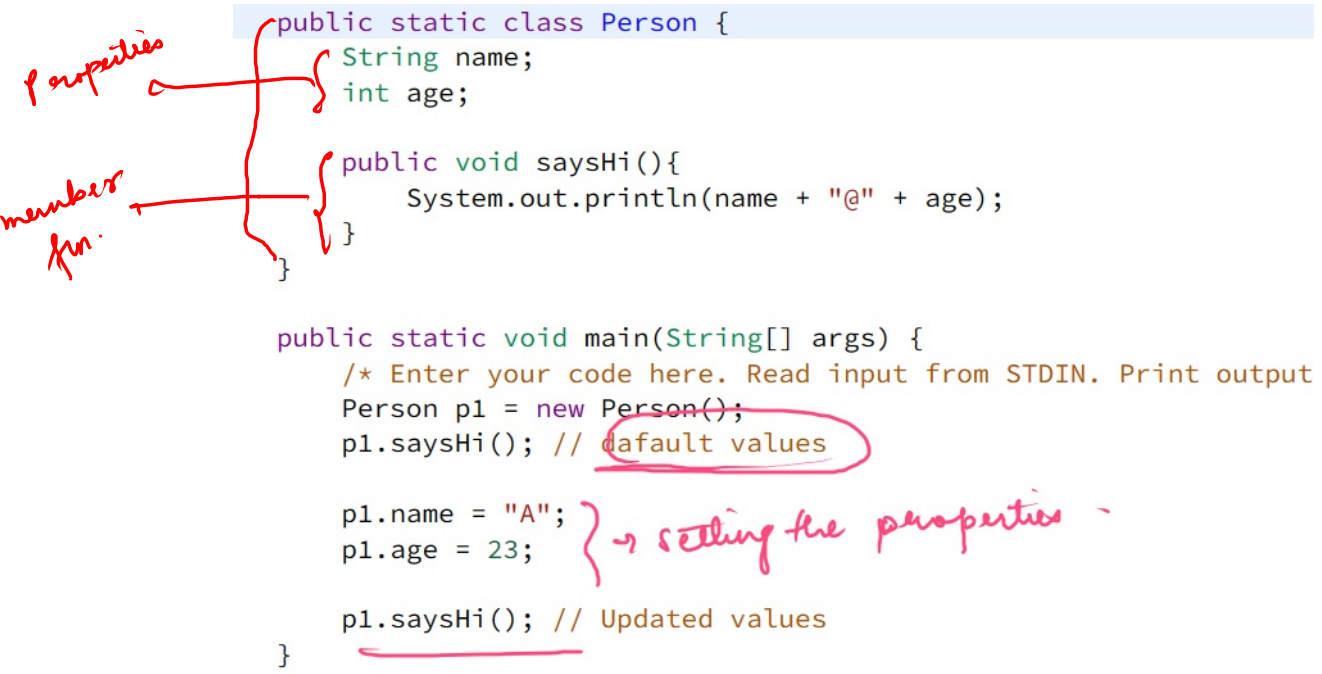
→ default values.

p1.name = "A"

p1.age = 23

A @ 23

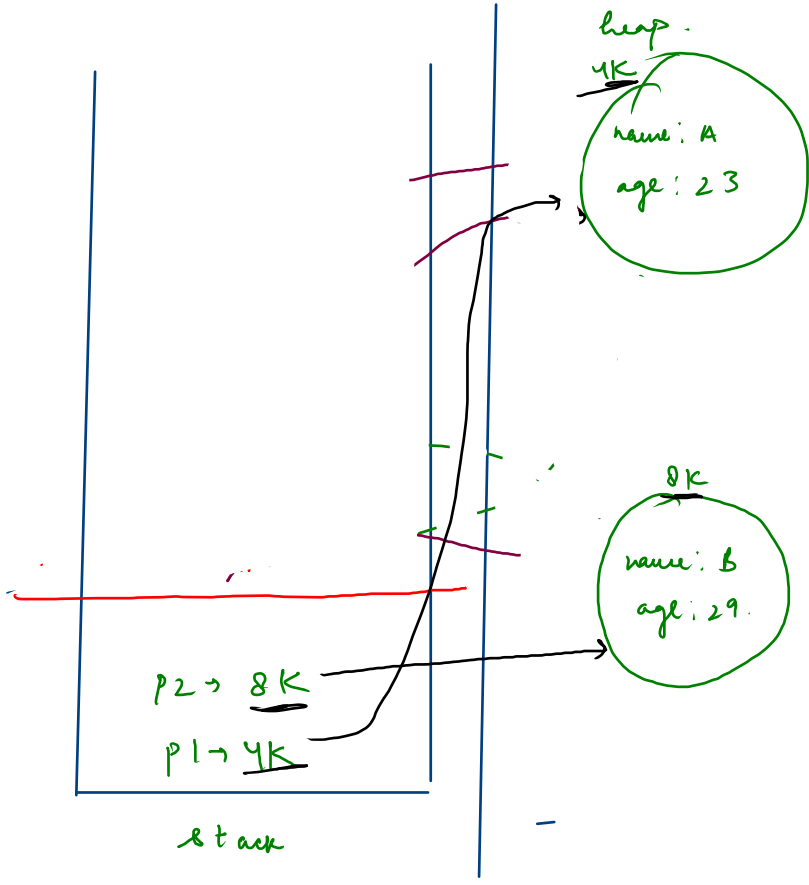



properties
member fun.
public static class Person {
String name;
int age;
public void saysHi(){
System.out.println(name + "@" + age);
}
}
public static void main(String[] args) {
/* Enter your code here. Read input from STDIN. Print output
Person p1 = new Person();
p1.saysHi(); // dafault values
p1.name = "A";
p1.age = 23;
→ setting the properties
p1.saysHi(); // Updated values
}

```java
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN.
    Person p1 = new Person();
    p1.name = "A";
    p1.age = 23;

    Person p2 = new Person();
    p2.name = "B";
    p2.age = 29;


    p1.saysHi();
    p2.saysHi();

    swap(p1,p2);

    p1.saysHi();
    p2.saysHi();
}
public static void swap(Person p1, Person p2){
    Person temp = p1;
    p1 = p2;
    p2 = temp;
}
```

p1.saysHi(); → A@23
p2.saysHi(); → B@29 instance

p1.saysHi(); → A@23
p2.saysHi(); → B@29



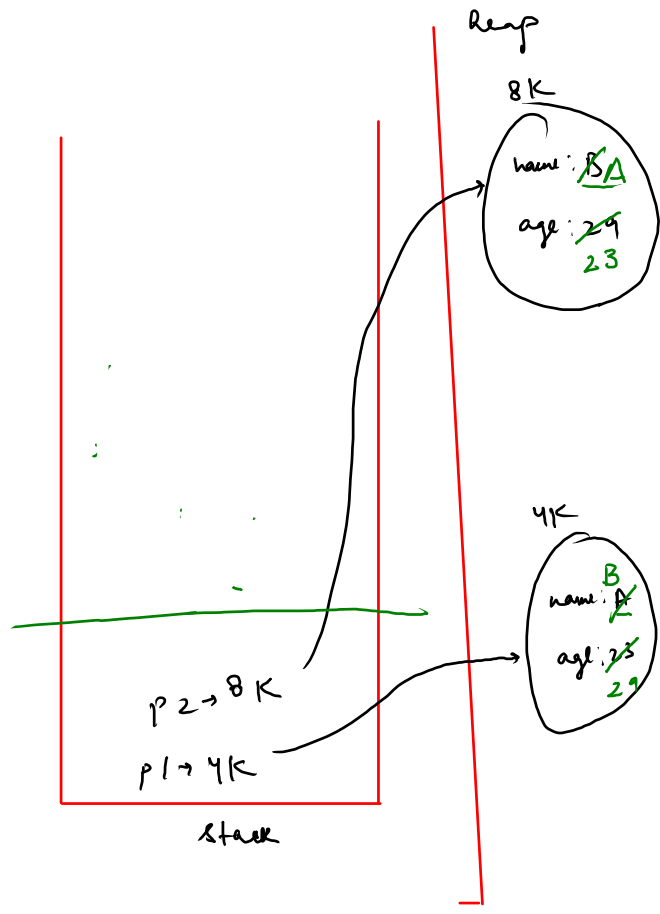
```
    /* Enter your code here. Read input from STDIN. Print ou
    Person p1 = new Person();
    p1.name = "A";
    p1.age = 23;

    Person p2 = new Person();
    p2.name = "B";
    p2.age = 29;


    p1.saysHi();
    p2.saysHi();

    swap(p1,p2);

    p1.saysHi();
    p2.saysHi();
}
public static void swap(Person p1, Person p2){}

public static void swap(Person p1, Person p2){
    String t = p1.name;
    p1.name = p2.name;
    p2.name = t;

    int temp = p1.age;
    p1.age  = p2.age;
    p2.age = temp;
}
```

→ A @ 23 → properties

→ B @ 29

→ B @ 29

→ A @ 23



```java
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print outpu
    Person p1 = new Person();
    p1.name = "A";
    p1.age = 23;

    Person p2 = new Person();
    p2.name = "B";
    p2.age = 29;


    p1.saysHi();
    p2.saysHi();

    swap(p1,p2);

    p1.saysHi();
    p2.saysHi();
}
public static void swap(Person p1, Person p2){}

public static void swap(Person p1, Person p2){}

public static void swap(Person p1, Person p2){
    String t = p1.name;
    p1.name = p2.name;
    p2.name = t;

    p1 = new Person();
    int temp = p1.age;
    p1.age = p2.age;
    p2.age = temp;
}
```

A @ 23
B @ 29

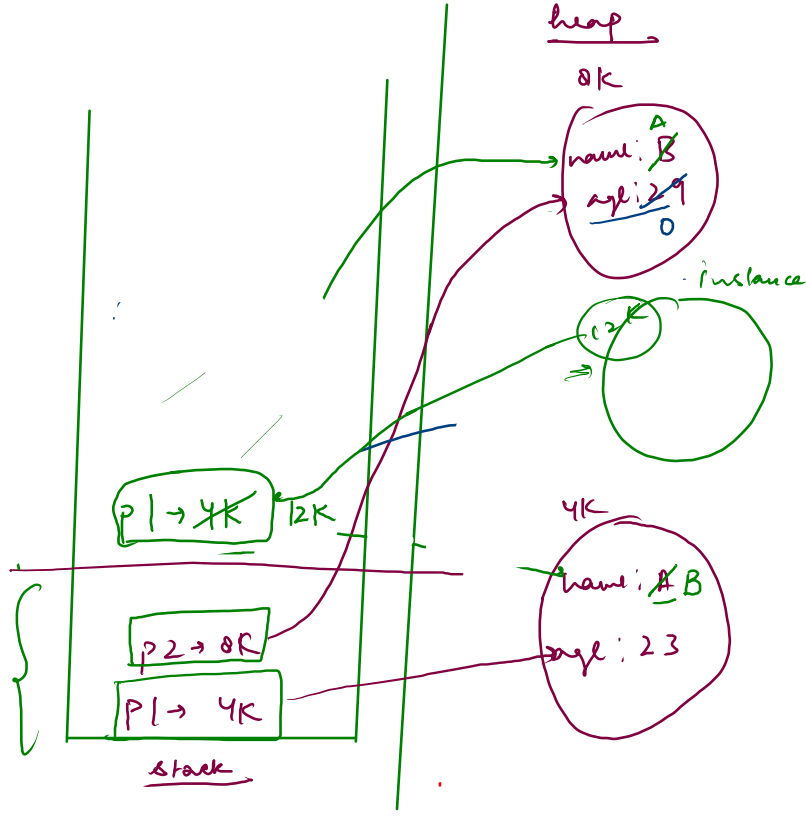
B @ 23
A @ 0

4K

creating new instance of p1 inside swap fn.

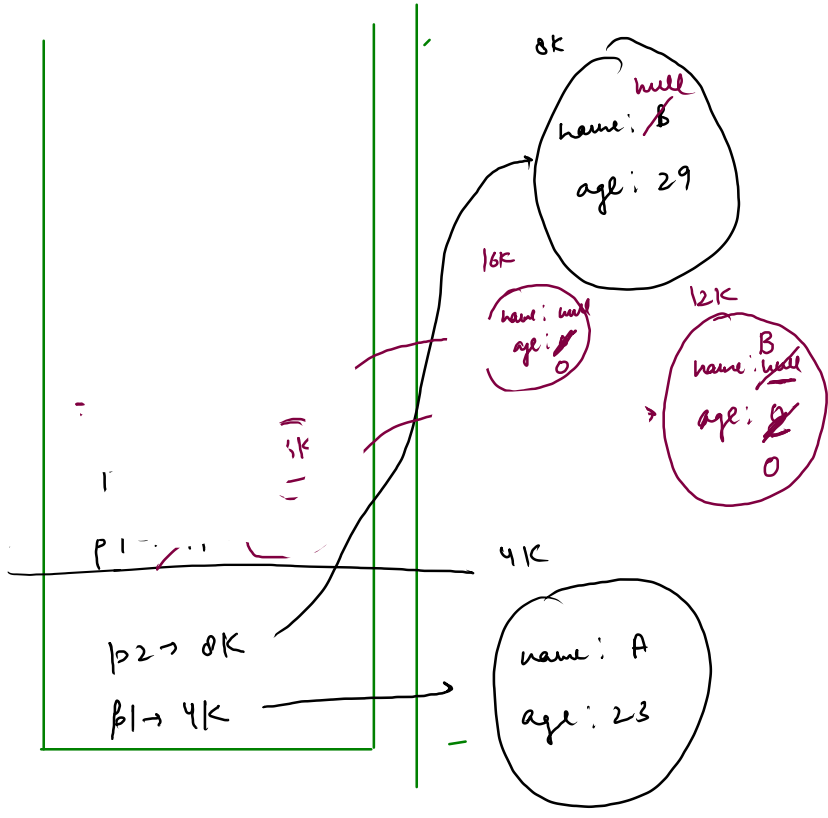
swap

main



```java
    Person p1 = new Person();
    p1.name = "A";
    p1.age = 23;

    Person p2 = new Person();
    p2.name = "B";
    p2.age = 29;


    p1.saysHi();
    p2.saysHi();

    swap(p1,p2);

    p1.saysHi();
    p2.saysHi();
}
public static void swap(Person p1, Person p2){}

public static void swap(Person p1, Person p2){}

public static void swap(Person p1, Person p2){}

public static void swap(Person p1, Person p2){
    p1 = new Person();
    String t = p1.name;
    p1.name = p2.name;
    p2.name = t;

    p2 = new Person();
    int temp = p1.age;
    p1.age = p2.age;
    p2.age = temp;
}
```

A @ 23
null @ 29



Constructor:-

```
public static class Person {
    String name;
    int age;

    public void saysHi(){
        System.out.println(name + "@" + age);
    }
}

public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT
    Person p1 = new Person();
    p1.name = "A";
    p1.age = 23;

    Person p2 = new Person();
    p2.name = "B";
    p2.age = 29;
```

↳ creating constructor.

↓

default

↓

Java.

↳ function with name similar to class name.

↳ default constructor is provided by Java.

↳ no return type even not (void) ✗

↳ initializing data members.

↳ Parameterized constructor.

↳ Created by us.

↳ Once we created a constructor, it is that Java doesn't provide us default constructor.

```
public class Solution {
    public static class Person {
        String name;
        int age;

        public void saysHi(){
            System.out.println(name + "@" + age);
        }

        public Person(){
            // default Constructor
        }

        public Person(String name, int age){
            Parametrized Constructor
            this.name = name;
            this.age = age;
        }

        public Person(String name){
            Parametrized Constructor
            this.name = name;

        }

        public Person( int Age){
            Parametrized Constructor
            age = Age;
        }
    }
```

```
    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print out
        Person p1 = new Person();
        p1.name = "A";
        p1.age = 23;

        Person p2 = new Person();
        p2.name = "B";
        p2.age = 29;


        Person p3 = new Person("C", 34);
        p3.saysHi();

        Person p4 = new Person("D");
        p4.saysHi();

        Person p5 = new Person(45);
        p5.saysHi();
    }
```