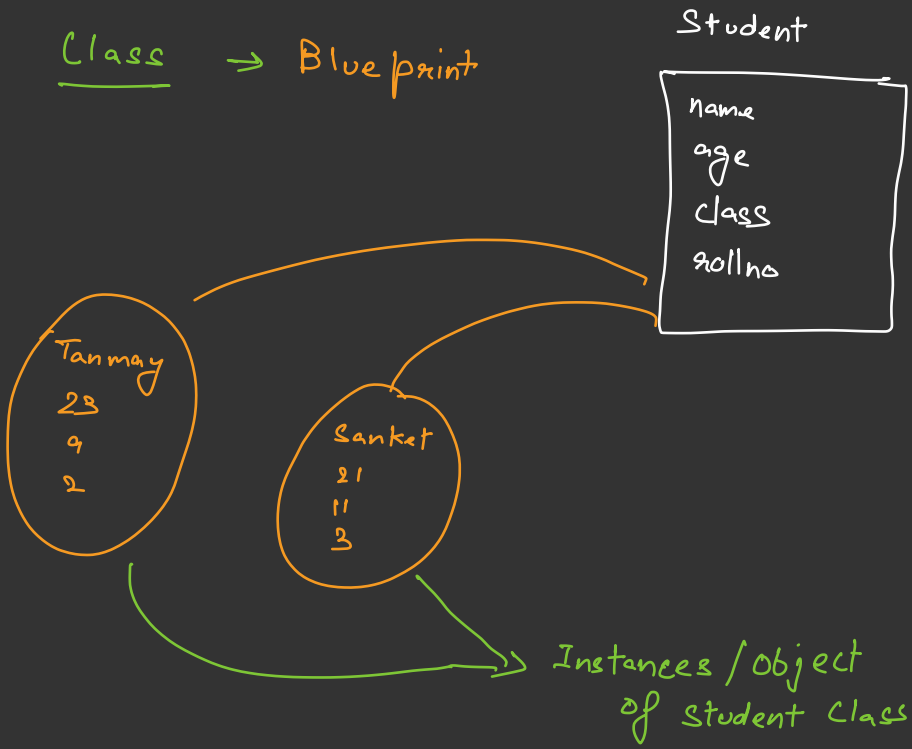
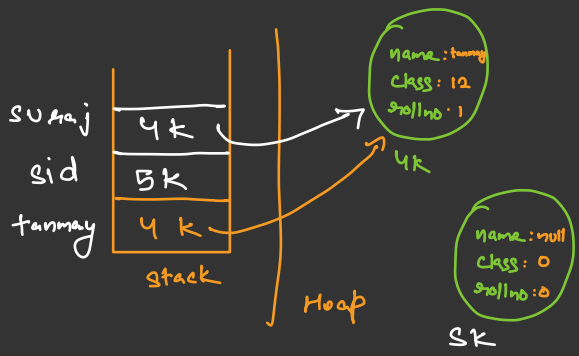
OOPs

Class → Blueprint




```
class Student {
    String name;
    int class;
    int rollNo;
}
```




```
main() {
    Student tanmay;
    tanmay = new Student();
    Student sid = new Student()
    Student suraj = tanmay;
```

. Operator

```
tanmay.name = "Tanmay";
tanmay.class = 12;
tanmay.rollNo = 1;
```

```
Syso(suraj.name)    // Tanmay
```

```
class Student {
    String name;
    int class;
    int rollNo;

    void printInfo {
        syso("Name : " + name);
        syso("class : " + class);
        syso("rollNo : " + rollNo);
```

this → gts a keyword that reference to current instance

```
void printInfo(){
    Syso( this.name)
}
```

name

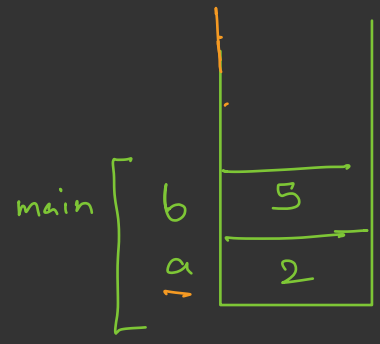
```
Student s = new Student()
s.printInfo();
```

Constructor

- special function
- name is same as that of a class
- No return type
- gets invoked when object is created.

```
p s  void swap (int a, int b)
{
    int t = a;
    a = b;
    b = t;
}

void main() {
    int a = 2;
    int b = 5;
    syso(a+" "+b);      // 2 5
    swap(a, b);
    syso(a+" "+b);      // 2 5
}
```



```
p s  void swap (int a [], int i, int j){
    int t = a [i];
    a [i] = a [j];
    a [j] = t;
}

main() {
    int arr = {1, 2, 3, 4}
    syso (arr [0] +" "+ arr [3]);   // 1 4
    swap (arr, 0, 3)
    syso (arr [0] + " " + arr [3];  // 4 1
```