

```
class student {
    static int no_of_students;
```

```
    static void fn();
```

```
}
```

```
main () {
```

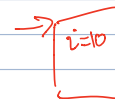
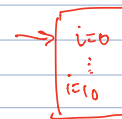
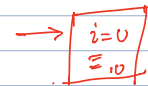
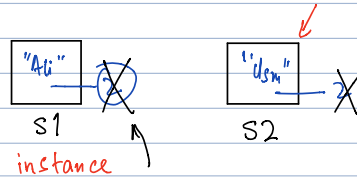
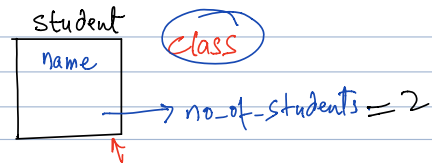
```
    student s1;
```

```
    //s1.no_of_students X
```

```
    student::no_of_students = 0;
```

scope resolution operator

```
    student::fn();
```



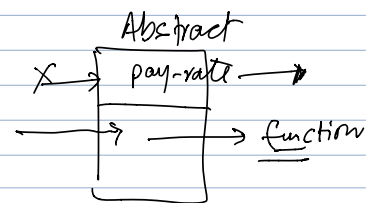
Static function locals = persist values between invocation.

```
class Employee {
    int pay_rate;
```

```
public:
```

```
void set_pay_rate(int rate) {
    if (rate > 14) {
        pay_rate = rate;
    }
}
```

```
}
```



"getters" and "setters"

public, protected, private → "access modifiers" —  
"access specifier" —

Abstraction

