

Classes

```
class Employee {
```

```
    id!: number;  
    name!: string;  
    address!: string;
```

! is used to avoid error that val. is undefined.

```
1) constructor() {}
```

You can only have one constructor at a time
1) default or 2) parameterized not both.

```
2) constructor(id: number, name: string, address: string)  
    {  
        this.address = address  
        this.id = id  
        this.name = name  
    }
```

```
    getNameWithAddress(): string {
```

```
        // return this.name + " " + this.address; // buggy use diff syntax  
        return ` ${this.name} stays at ${this.address}`
```

```
    }
```

Access modifier

```
#id    } private (only within class)  
private id
```

protected name (within class, and in classes that extend it)

Static members

```
static getCount(): number {  
    return 50;  
}
```

```
Employee.getCount();
```

// static members can be accessed with
class name.

Getters and Setters

```
get empId(): number  
{  
    return this.id  
}
```

```
set empId(id: number) {
```

`this.id = id`
}