

```
class Employee {
```

```
public:
```

```
string name;
```

```
int salary;
```

```
void sign-in();
```

members / fields

methods

✓ Employee() → class constructor

✓ Employee(int salary);
~Employee();
↳ "tilda"

};

function/method

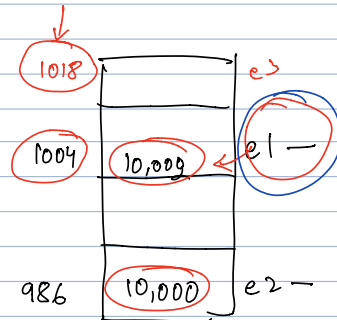
same name
num/type of
parameters diff.

"method overloading".

```
Employee::Employee() {
    salary = 10000;
    cout << "Employee created" << endl;
}
```

↳ "logic"

↳ field/member



```
Employee::Employee(int salary) {
    this->salary = 70,000;
}
```

this → salary = 70,000

field, "not" local → local

```
Employee::Employee(int s) {
    this->salary = s;
}
```

```
main() {
```

```
Employee
```

```
e1;
```

```
cout << e1.salary << endl;
```

```
e1.salary = 10000;
```

```
Employee e2;
```

```
cout << e2.salary;
```

```
Employee e3(70000);
```

```
cout << e3.salary;
```

←

←

←

← 10,000