let savingsAccount = {

"nayeem"

Bank Name: "Axis"

Account no: "---"

Account Balace:

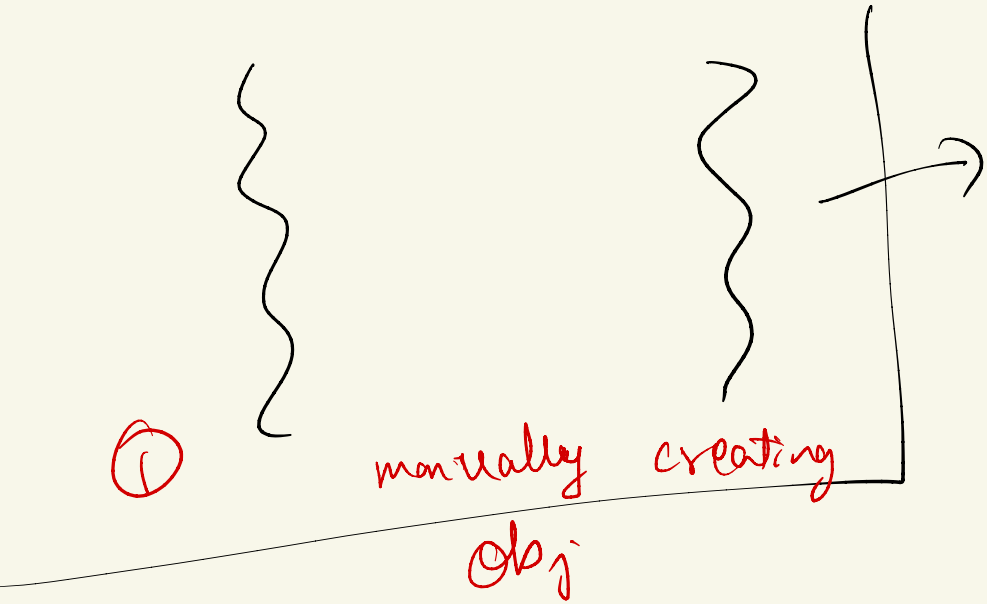
withdraw: ==

deposit : =

}

```
let user = {

    name: "Sujash"

    accountBalance: ''

    BankName: ''

    addMoney: () {}

}
```

```
function createCustomer(name, balance,
    bankName) {

    let user = { };

    user.name = name;
    user.balance = balance;
    user.bankname = bankName
    user.addMone =>
    return user;

}
```



creating user Obj through fn

let user = {

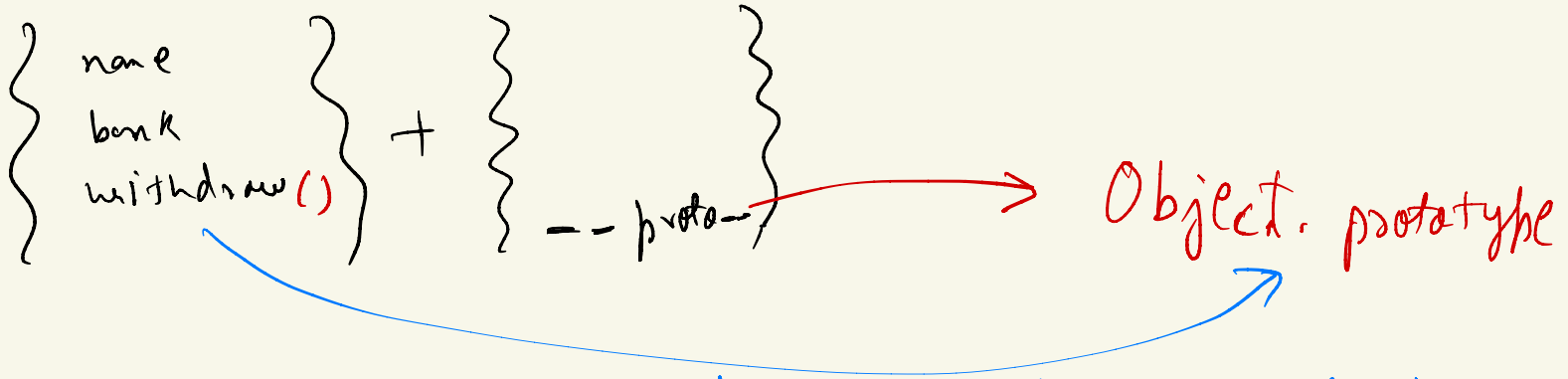
name →

bankName →

accountBalance →

withdraw → { }

}



* we will be able to solve memory issue*.

* createCustomer → const fn.

---

function createCustomer () {

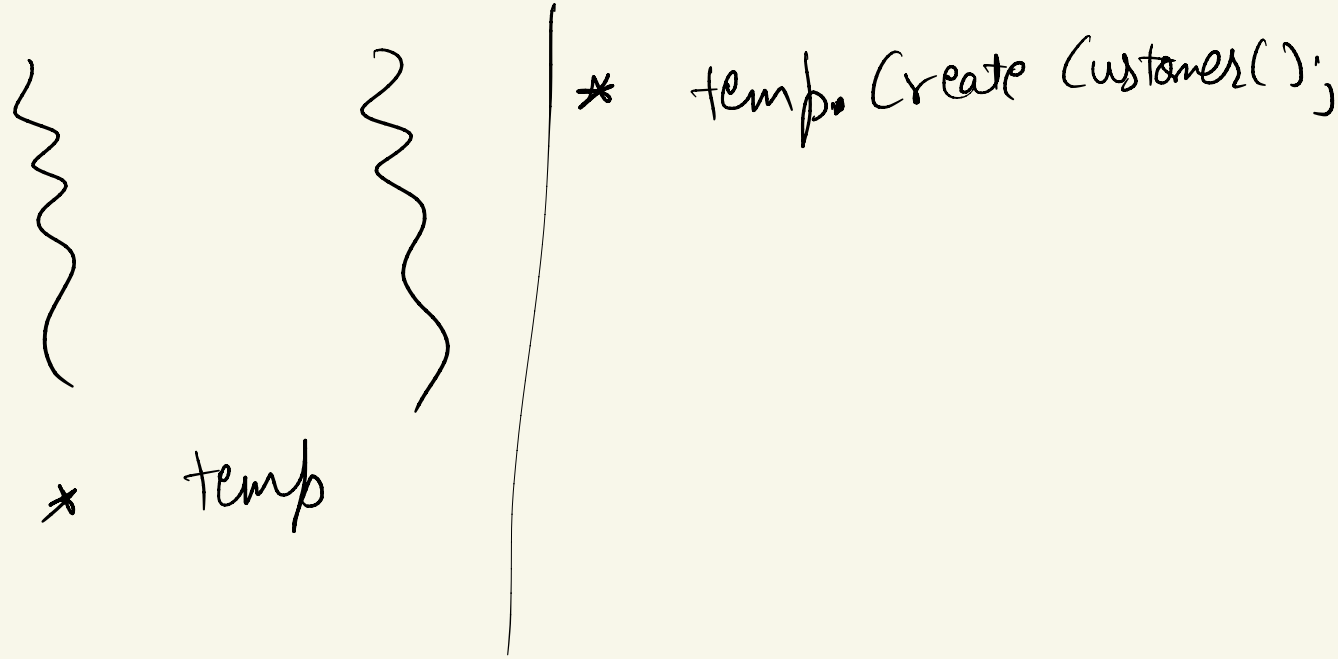
}

let user = new createCustomer ();

Creates a new memory

* new keyword → creates a new memory.
* createCustomer is called with ref of new memory.

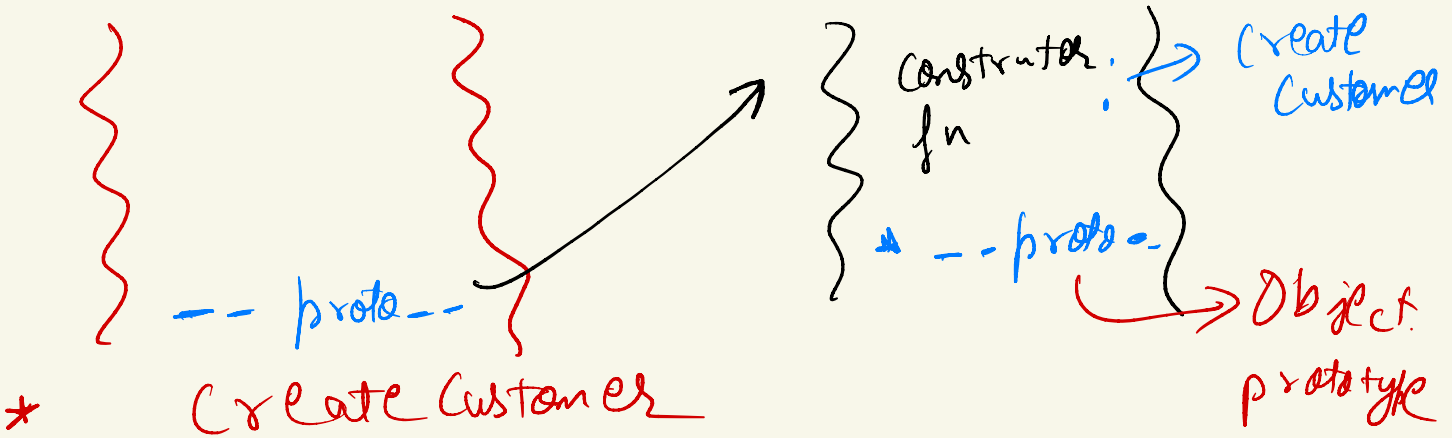
* let user = new Createcustomer ();



constructorFn.js:30

```
▼ CreateCustomer {name: 'naveen', bankName: 'axis', balance: 5000} i
    balance: 5000
    bankName: "axis"
    name: "naveen"
  ▼ [[Prototype]]: Object
    ▶ constructor: f CreateCustomer(name, bankName, balance)
    ▶ [[Prototype]]: Object
```

* let user = new CreateCustomer ();

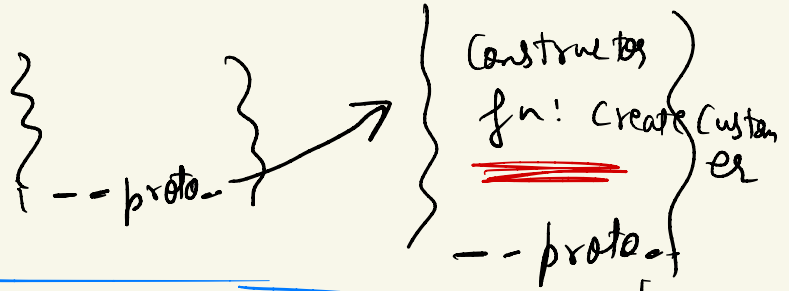


method 1 , method 2

Object

user 1, user 2

method 3 → new key word



Object

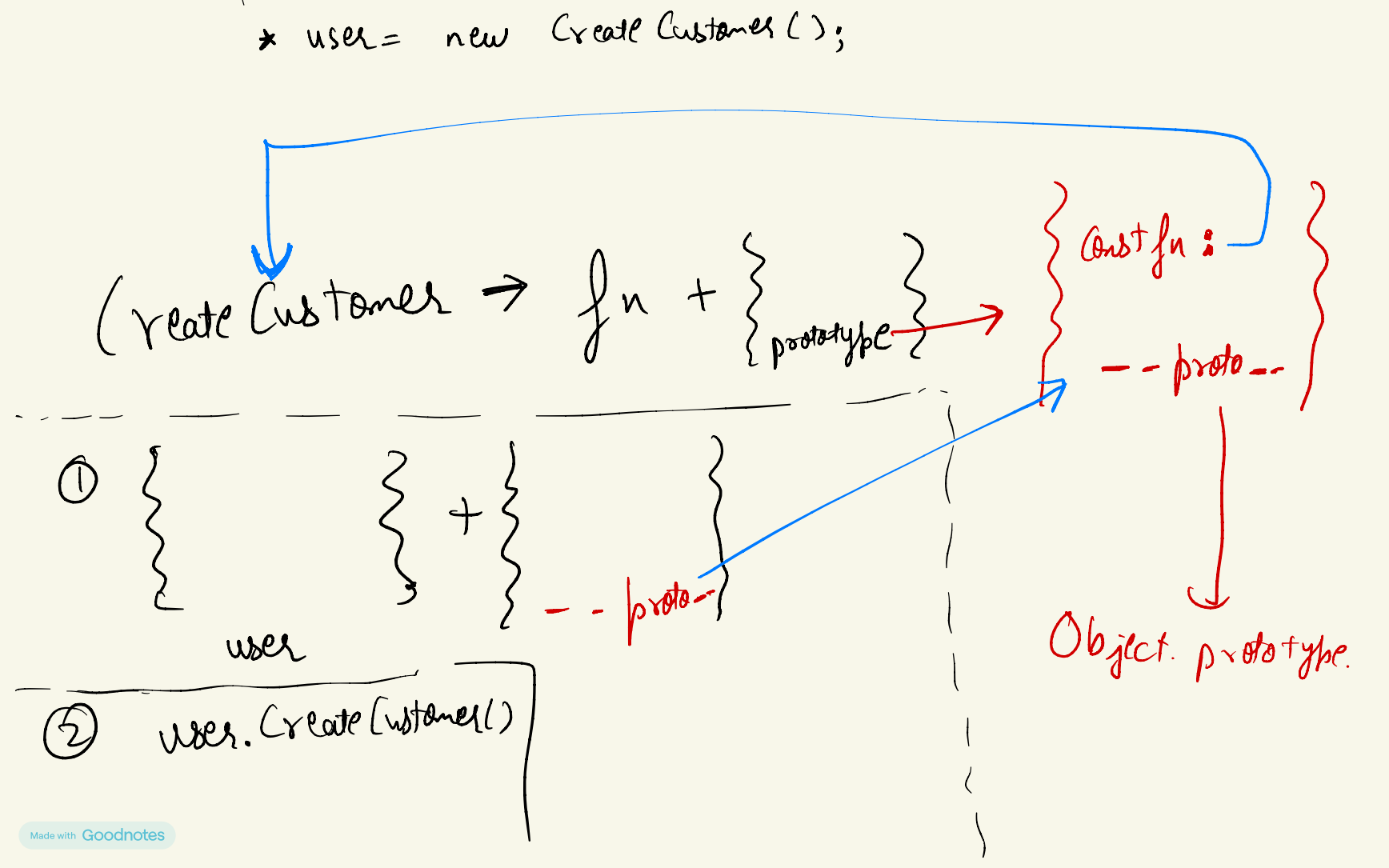
↓

createCustomer

↓

user 3

* user = new CreateCustomer();



① new keyword creates a new memory (user)

② Create Customer is executed with the ref of new memory created.

③ --proto-- of new memory created points to Create Customer.prototype.