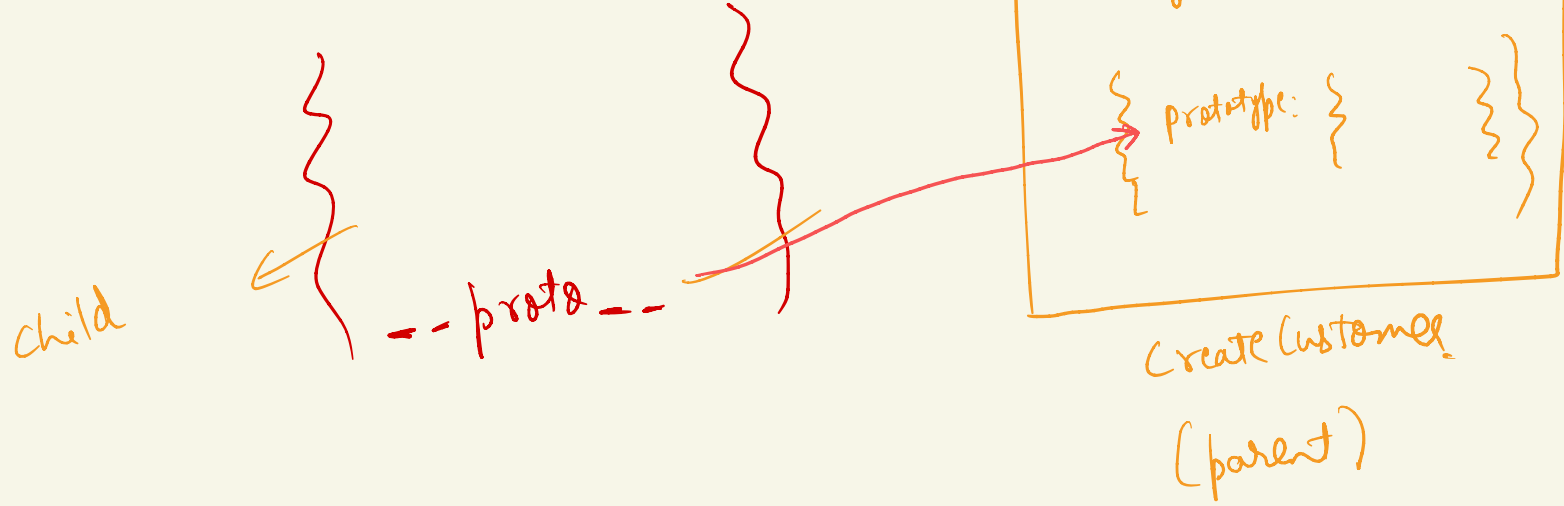



createCustomer \longrightarrow fn
+ { prototype : { storing
hidden
methods } }

\Rightarrow createCustomer() { this will behave
as function }
 \Rightarrow createCustomer.prototype

→ Customer1 is instance of createCustomer



constructorFunctions.js:7

```
▼ createCustomer {} ⓘ  
  ▼ [[Prototype]]: Object  
    ▶ constructor: f createCustomer(nam, branch, accountBalance)  
  ▼ [[Prototype]]: Object  
    ▶ constructor: f Object()  
    ▶ hasOwnProperty: f hasOwnProperty()  
    ▶ isPrototypeOf: f isPrototypeOf()  
    ▶ propertyIsEnumerable: f propertyIsEnumerable()  
    ▶ toLocaleString: f toLocaleString()  
    ▶ toString: f toString()  
    ▶ valueOf: f valueOf()  
    ▶ __defineGetter__: f __defineGetter__()  
    ▶ __defineSetter__: f __defineSetter__()  
    ▶ __lookupGetter__: f __lookupGetter__()  
    ▶ __lookupSetter__: f __lookupSetter__()  
    ▶ __proto__: (...)  
    ▶ get __proto__: f __proto__()  
    ▶ set __proto__: f __proto__()
```

→ Child of
Create Customer

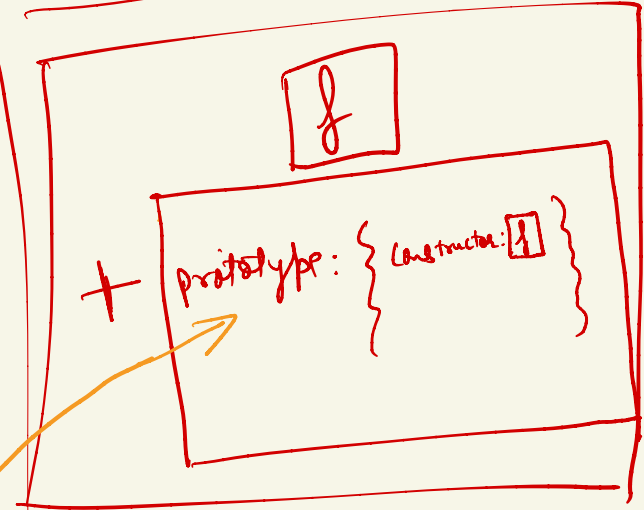
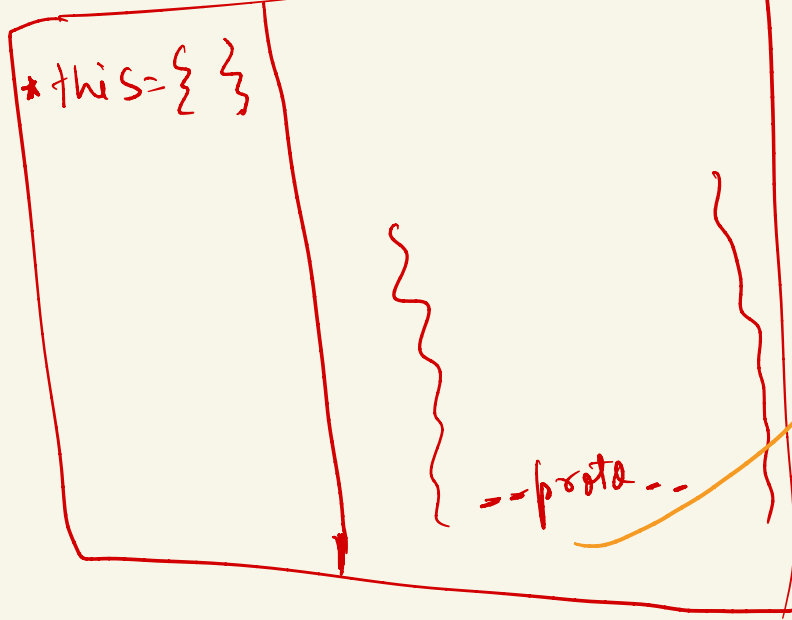
constructorFunctions.js:15

```
▼ {} ⓘ  
  ▼ [[Prototype]]: Object  
    ▶ constructor: f Object()  
    ▶ hasOwnProperty: f hasOwnProperty()  
    ▶ isPrototypeOf: f isPrototypeOf()  
    ▶ propertyIsEnumerable: f propertyIsEnumerable()  
    ▶ toLocaleString: f toLocaleString()  
    ▶ toString: f toString()  
    ▶ valueOf: f valueOf()  
    ▶ __defineGetter__: f __defineGetter__()  
    ▶ __defineSetter__: f __defineSetter__()  
    ▶ __lookupGetter__: f __lookupGetter__()  
    ▶ __lookupSetter__: f __lookupSetter__()  
    ▶ __proto__: (...)  
    ▶ get __proto__: f __proto__()  
    ▶ set __proto__: f __proto__()
```

→ Child of
obj

* Customer1 = new createCustomer (name, balance, branch); Global memory

local E.C



Create Customer

```
customer1 = new createCustomer (name, balance, branch);
```

no need
to write
both lines
as it is happening
because of new keyword

```
function createCustomer (name) {
```

```
  this = {} (createCustomer)  
            {type of obj}
```

```
  this.name = name;
```

```
  return this; (happened because  
               of new keyword)
```

```
}
```

You, 2 minutes ago | 1 author (you)

```
function createCustomer(nam, branch, accountBalance){  
  this.nam=nam;  
  this.branch=branch;  
  this.accountBalance=accountBalance;  
  this.addMoney = function(amount){  
    this.accountBalance+=amount;  
  }  
}
```

this: {
 nam: nam
 branch: branch
 accountBalance: —
 addMoney: function
 --proto--: }

CreateCustomer =

{ +

prototype: {
 constructor:
 : f
 —
 —
 —
 —
}

