Encapsulation

It refers to the practice of bundling data (attributes or properties) and methods (functions) that operate on that data into a single unit, known as "classes". Encapsulation helps in organizing and managing code by restricting direct access to internal data and providing controlled access through methods.

Abstraction



Abstraction in Constructor Functions

function BankAccount(balance)

{

    // Private Property

    let \_balance = balance;

    // Private Method

    let checkSufficientBalance = function(amount)

    {

        return amount <= \_balance;

    }

    this.withdraw = function(amount)

    {

        const isSufficientBalance = checkSufficientBalance(amount);

        if(isSufficientBalance)

        {

           \_balance -= amount;

           return;

        }

        console.log("Insufficient funds");

    }

    this.deposit = function(amount)

    {

        \_balance += amount;

    }

    this.getBalance = function()

    {

        return \_balance;

    }

}

const account = new BankAccount(1000);

account.deposit(500);

account.withdraw(200);

console.log(account.getBalance());  // Output: 1300

Abstraction in Classes

const \_balance = new WeakMap();

const \_checkSufficientBalance = new WeakMap();

class BankAccount

{

    constructor(balance)

    {

        \_balance.set(this, balance);

        \_checkSufficientBalance.set(this, (amount)=>

        {

            let balance = \_balance.get(this);

            return amount <= balance;

        });

    }

    getBalance()

    {

        return \_balance.get(this);

    }

    deposit(amount)

    {

        let balance = \_balance.get(this);

        \_balance.set(this, balance + amount);

    }

    withdraw(amount)

    {

        const isSufficientBalance = \_checkSufficientBalance.get(this)(amount);

        if(isSufficientBalance)

        {

            let balance = \_balance.get(this);

            \_balance.set(this, balance - amount);

            return;

        }

        console.log("Insufficient Balance");

    }

}

const account = new BankAccount(1000);

account.deposit(500);

account.withdraw(200);

console.log(account.getBalance());  // Output: 1300

account.withdraw(2000); //Insufficient Balance