**01 Getting Started**

**1) What is object-oriented programming**:

Object-oriented programming is a programming paradigm or style of programming that is centered around objects rather than functions. OOP is not a programming or tool it’s a style of programming or programming paradigm.

There are several programming languages that support object-oriented programming. For example, C#, Java, Python, JavaScript etc.

**2) Four Pillars of OOP**:

In object-oriented programming there are four core concepts. They are

1. Encapsulation
2. Abstraction
3. Inheritance
4. Polymorphism

In object-oriented programming we combined a group of related variables and functions into a unite. We call that unite an object. We refer these variables as property’s and functions as methods.

**a) Encapsulation**:

In object-oriented programming we group related variables and functions that are operate on them into objects. And this is called encapsulation. Encapsulation means binding code in a single unite.

**Example**:

*//functional programming*

let baseSalary = 3000;

let overtime = 10;

let rate = 20;

function getWage(baseSalary, overtime, rate) {

return baseSalary + overtime \* rate;

}

const total = getWage(baseSalary, overtime, rate);

console.log(total); *//3200*

*//Object-orianted programming*

let employee = {

baseSalary: 3000,

overtime: 10,

rate: 20,

getWage: function() {

return *this*.baseSalary + *this*.overtime \* *this*.rate;

}

};

console.log(employee.getWage()); *//3200*

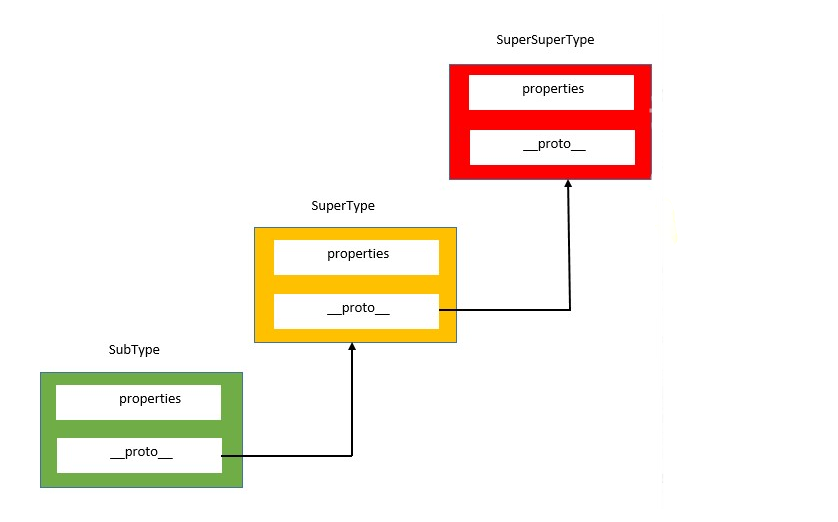
**b) Abstraction**:

In object-oriented programming abstraction means hide details implementation. All the complexity is hidden from outside from our object. And this gives us a couple of benefits, such as make the interface as simple, understanding the object easily, reduce the impact of change.

**c) Inheritance**:

Inheritance is a mechanism that allows us to aluminate redundant code. By inheritance we can make parent property and method available in child.

JavaScript does not have classes unlike other languages. It uses the concept of prototypes and prototype chaining for inheritance.



**d) Polymorphism**:

Poly means many and morphism means form. So, polymorphism means many same entities but different form. The ability to call the same method on different objects and have each of them respond in their own way is called polymorphism.

So, Using Encapsulation we group related variables and functions together and this way we can reduce complexity. Now we reuse the object in different part of the program.

Using Abstraction, we can hide the details and the complexity and show only the essential. This technique reduces complexity and also isolate the impact of changes in the code.

Using Inheritance, we can eliminate redundant code.

And with polymorphism we can refactor ugly switch/case statement.

**3) Setting Up the Development Environment**:

We have to install the following things for create our working environment.

1. Visual Studio Code
2. Live server extension

01 Getting Started