**Object Orientated Programs(OOP) in JS:**

**What is object-oriented programming (OOP) in JavaScript?**

Object-oriented programming (OOP) is a programming paradigm that involves defining classes and objects to represent real-world concepts and entities. In JavaScript, OOP involves creating objects that encapsulate data and functionality and then using those objects to create larger programs. OOP in JavaScript is based on the concept of prototypes, which allow objects to inherit properties and methods from other objects.

**What is a class in JavaScript?**

A class in JavaScript is a blueprint for creating objects that share the same properties and methods. A class defines the characteristics and behaviors of the objects it creates, but does not actually create any objects itself. To create an object from a class, you use the new keyword followed by the class name and any arguments required by the constructor.

**How do you create an object in JavaScript?**

There are several ways to create an object in JavaScript, but the most common method is to use object literal notation. Object literal notation involves defining an object using curly braces, with key-value pairs for each property and method. For example:

const myObject = {

name: 'John',

age: 30,

sayHello: function() {

console.log('Hello!');

}

};

**What is a constructor in JavaScript?**

A constructor in JavaScript is a special method that is used to create and initialize an object created from a class. The constructor is called when a new object is created, and it is responsible for setting the initial state of the object by setting its properties and calling any necessary methods.

**What is inheritance in JavaScript?**

Inheritance in JavaScript is the ability for an object to inherit properties and methods from another object. In JavaScript, inheritance is implemented using prototypes. When an object is created, it inherits properties and methods from its prototype object, which in turn may inherit properties and methods from its own prototype object. This creates a chain of objects that inherit from one another.

**What is the prototype in JavaScript?**

The prototype in JavaScript is an object that is used to provide inheritance between objects. Every object in JavaScript has a prototype object, which is used to look up properties and methods that are not defined on the object itself. When a property or method is accessed on an object, JavaScript first checks to see if it is defined on the object itself. If it is not, it looks for it on the object's prototype, and continues up the prototype chain until it is found or the end of the chain is reached.

**How do you define a prototype in JavaScript?**

To define a prototype in JavaScript, you can use the Object.create() method. This method takes an object as its argument, and creates a new object that inherits from the passed-in object. For example:

const myPrototype = {

name: 'John',

age: 30,

sayHello: function() {

console.log('Hello!');

}

};

const myObject = Object.create(myPrototype);

**What is the difference between classical inheritance and prototypal inheritance?**

Classical inheritance is the traditional method of implementing inheritance in object-oriented programming, where classes are used to create objects, and objects inherit from their parent classes. Prototypal inheritance, on the other hand, is a form of inheritance where objects inherit directly from other objects, without the need for classes. In JavaScript, prototypal inheritance is used instead of classical inheritance.