**OOPS=>**

Object-Oriented Programming (OOP) in JavaScript is a programming paradigm that uses objects and classes to structure code and represent real-world concepts. OOP allows developers to create reusable, modular, and maintainable code by organizing data and behaviors into objects and using inheritance and other OOP concepts.

**Key Concepts in OOP**

1. **Objects**: Objects are collections of related data and functions (methods). They represent entities with attributes (properties) and behaviors (methods).
2. **Classes**: Classes are blueprints for creating objects. They define properties and methods that objects of that class will have.
3. **Encapsulation**: Encapsulation involves bundling data and methods that operate on that data within an object, restricting direct access to certain components.
4. **Inheritance**: Inheritance allows a class to inherit properties and methods from another class, promoting code reuse and establishing a relationship between classes.
5. **Polymorphism**: Polymorphism allows objects to be treated as instances of their parent class, enabling different implementations of the same method.

**Classes and Constructors**

Classes are a way to create objects based on a blueprint. You can define a constructor function that initializes object properties and methods:

class Person {

constructor(name, age) {

this.name = name;

this.age = age;

}

greet() {

console.log(`Hello, my name is ${this.name}.`);

}

}

const john = new Person("John", 30);

john.greet(); // Output: "Hello, my name is John."

**Inheritance and Subclasses**

Inheritance allows one class to inherit properties and methods from another. The subclass (child class) inherits from the superclass (parent class):

class Employee extends Person {

constructor(name, age, position) {

super(name, age); // Call the parent class constructor

this.position = position;

}

work() {

console.log(`${this.name} is working as a ${this.position}.`);

}

}

const jane = new Employee("Jane", 28, "Engineer");

jane.greet(); // Output: "Hello, my name is Jane."

jane.work(); // Output: "Jane is working as an Engineer."