Q.1 What’s Constructor And Its Purpose?

A constructor is a special method in a class that is called when an object of that class is created using the `new` keyword. It is used to initialize the object's state and define its initial properties and values. The purpose of a constructor is to set up the object's initial state and perform any necessary setup or initialization tasks.

Q.2 Explain This Keyword and Its Purpose?

The `this` keyword refers to the current instance of an object and is used within methods or constructors of a class to access or refer to the object's properties and methods. It allows you to access and manipulate the current object's state and behavior. The purpose of the `this` keyword is to provide a way to refer to the specific instance of the object that is executing the current code block.

Q.3 What’s Call Apply Bind Method & Difference Between them?

The `call`, `apply`, and `bind` are methods in JavaScript used to manipulate the `this` value and invoke functions with a specified context. The main differences are as follows:

- `call`: The `call` method is used to invoke a function with a specified `this` value and individual arguments provided as separate arguments to the function. It allows you to pass arguments directly as a comma-separated list.

- `apply`: The `apply` method is similar to `call`, but it takes the `this` value as the first argument and an array or an array-like object containing the arguments to be passed to the function.

- `bind`: The `bind` method creates a new function with a bound `this` value and optionally prepends arguments to the original function. It allows you to create a new function with a permanently bound `this` value, which can be invoked later.

Q.4 Explain OOPS?

OOPS (Object-Oriented Programming) is a programming paradigm that organizes code into objects that interact with each other. It emphasizes the concepts of encapsulation, inheritance, and polymorphism. OOPS aims to structure code in a way that models real-world entities and behaviors, making it easier to understand, maintain, and extend the codebase.

Q.5 Whats Abstraction and Its Purpose?

Abstraction is a fundamental concept in OOPS that focuses on hiding unnecessary implementation details and exposing only the essential features and behavior. It allows you to create abstract classes or interfaces that define a blueprint for objects, without specifying the exact implementation. The purpose of abstraction is to simplify complex systems by providing a clear and concise interface for users while hiding the underlying complexity.

Q.6 Whats Polymorphism and Purpose of it?

Polymorphism refers to the ability of objects to have multiple forms or behaviors. It allows objects of different classes to be treated as objects of a common superclass or interface. The purpose of polymorphism is to create code that can work with objects of various types, enabling more flexibility and extensibility in the codebase.

Q.7 What's Inheritance and Purpose of it?

Inheritance is a concept in OOPS that allows a class to inherit properties and methods from another class. It establishes a parent-child relationship between classes, where the child class inherits the characteristics of the parent class. The purpose of inheritance is to promote code reusability, as common attributes and behaviors can be defined in a parent class and shared by multiple derived (child) classes.

Q.8 What's Encapsulation and Purpose of it?

Encapsulation is a principle in OOPS that bundles data and methods (functions) together within a class, hiding the internal state and implementation details from the outside world. It provides a protective barrier around the data, allowing controlled access through methods. The purpose of encapsulation is to achieve data abstraction, maintain data integrity, and control access to the internal state of an object.

Q.9 Explain Class in JavaScript?

A class in JavaScript is a template or blueprint for creating objects. It defines the structure, properties, and methods that objects instantiated from the class will have. It provides a way to define objects with similar characteristics and behaviors. Classes in JavaScript were introduced in ECMAScript 2015 (ES6) and offer syntactic sugar over the traditional prototype-based inheritance model.

Q.10 What's Super Keyword & What it does?

The `super` keyword in JavaScript is used inside a subclass constructor to refer to the parent class and call its constructor or methods. It allows the subclass to access and invoke the parent class's constructor and properties. The `super` keyword is typically used in the constructor of the subclass to ensure that the parent class's initialization logic is executed before adding additional functionality to the subclass.