PPT Program Assignment

Web Development Assignment - 6

Answer 1: A constructor in JavaScript is a special method used to create and initialize objects. It sets up the initial state and behavior of an object. Constructors ensure consistent object creation and allow customization through parameters.

Answer 2: The ‘this’ keyword in JavaScript refers to the current execution context or the object that the current function belongs to. Its purpose is to access and manipulate properties and methods within an object or class.

When used inside a method of an object, ‘this’ allows you to refer to other properties or methods within the same object. It enables the object to reference itself and access its own members.

The value of ‘this’ is determined dynamically at runtime based on how a function is called. It can change depending on the context in which the function is invoked, such as when using event handlers or callback functions.

Answer 3: The call method invokes a function with a specified ‘this’ value and individual arguments.
The ‘apply’ method invokes a function with a specified ‘this’ value and arguments passed as an array or array-like object.
The ‘bind’ method creates a new function with a bound ‘this’ value and optional pre-set arguments, returning a function that can be invoked later.
‘call’ and ‘apply’ immediately invoke the function, while ‘bind’ returns a new function for later invocation.
‘call’ and ‘apply’ accept arguments directly, while ‘apply’ accepts arguments as an array-like object.
‘bind’ allows you to create a partially applied function with a fixed ‘this’ value and pre-set arguments.

Answer 4: Object-Oriented Programming (OOP) is a programming paradigm that organizes code around objects, which are instances of classes. It focuses on encapsulating data and behavior into reusable and modular units. OOP principles include encapsulation, inheritance, and polymorphism, allowing for code reusability, modularity, and easier maintenance. It promotes a structured and organized approach to software development, enhancing code readability and scalability.

Answer 5: Abstraction is a fundamental concept in object-oriented programming that focuses on hiding complex implementation details and exposing only relevant information and functionality to the outside world. Its purpose is to simplify the usage and understanding of a system or a class by providing a simplified and well-defined interface. Abstraction allows developers to create higher-level concepts and models, making code more modular, reusable, and maintainable.

Answer 6: Polymorphism is a concept in object-oriented programming that allows objects of different classes to be treated as objects of a common superclass. It enables a single interface to be implemented by multiple classes, each providing their own implementation of the interface methods. The purpose of polymorphism is to simplify code by promoting code reuse, flexibility, and extensibility. It allows for writing generic code that can handle objects of various types, enhancing code modularity and supporting dynamic behavior based on the actual object type at runtime.

Answer 7: Inheritance is a fundamental concept in object-oriented programming where a class can inherit properties and methods from another class. It allows for creating a hierarchy of classes, with subclasses inheriting characteristics of a superclass. The purpose of inheritance is to promote code reuse, eliminate redundancy, and establish an "is-a" relationship between classes. It enables the implementation of a general class that can be specialized by more specific subclasses, enhancing code organization, and supporting modular and scalable software development.

Answer 8: Encapsulation is a principle in object-oriented programming that involves bundling data and methods together within a class and controlling access to them. Its purpose is to protect the internal state of an object, prevent direct manipulation of data, and ensure that interactions with the object occur through defined interfaces. Encapsulation promotes code maintainability, reusability, and reduces the likelihood of conflicts or unintended modifications, providing a clear separation between implementation details and external interactions.

Answer 9: In JavaScript, a class is a blueprint or template for creating objects that share similar properties and behaviors. It serves as a template for defining object structure, data, and methods. With the introduction of ES6, JavaScript has a class syntax that simplifies the process of creating objects and implementing object-oriented concepts such as inheritance and encapsulation.

Answer 10: In JavaScript, the super keyword is used within a subclass to call the parent class's constructor or access its methods and properties. It provides a way to refer to the parent class and invoke its functionalities. The 'super' keyword is typically used in the constructor of a subclass to pass arguments to the parent class's constructor using 'super()' or to call parent class methods using 'super.methodName()'. It enables the subclass to inherit and extend the behavior of the parent class while maintaining access to its functionality.