Q.1 What’s Constructor And Its Purpose?

In Java, Constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling the constructor, memory for the object is allocated in the memory. It is a special type of method that is used to initialize the object. Every time an object is created using the new() keyword, at least one constructor is called.

Constructor Types

1. Default Constructor
2. Parameterized Constructor
3. Copy Constructor
4. Static Constructor
5. Private Constructor

Q.2 Explain This Keyword and Its Purpose?

The most common use of the this keyword is to eliminate the confusion between class attributes and parameters with the same name (because a class attribute is shadowed by a method or constructor parameter). If you omit the keyword in the example above, the output would be "0" instead of "5".

This can also be used to:

- Invoke current class constructor
- Invoke current class method
- Return the current class object
- Pass an argument in the method call
- Pass an argument in the constructor call

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

Call:- The **call method** binds the this value to the function and executes the function. It takes the this value and a list of arguments as parameters. Then, it returns the value returned by the function, which is called using the call method.

Apply:- The apply method binds the this value to the function and executes the function. It takes the this value and a single array object as parameters, and it returns the value returned by the function, which is called using the apply method.

Bind:-The **bind method** binds the this value to the function and returns a new function. However, we still need to separately invoke the returned function.

Q.4 Explain OOPS ?

Object-oriented design started right from the moment computers were invented. Programming was there, and programming approaches came into the picture. Programming is basically giving certain instructions to the computer.
At the beginning of the computing era, programming was usually limited to machine language programming. Machine language means those sets of instructions that are specific to a particular machine or processor, which are in the form of 0’s and 1’s. These are sequences of bits (0100110…). But it’s quite difficult to write a program or develop software in machine language.
It’s actually impossible to develop software used in today’s scenarios with sequences of bits. This was the main reason programmers moved on to the next generation of programming languages, developing assembly languages, which were near enough to the English language to easily understand. These assembly languages were used in microprocessors. With the invention of the microprocessor, assembly languages flourished and ruled over the industry, but it was not enough. Again, programmers came up with something new, i.e., structured and procedural programming.

Q.5 Whats Abstraction and Its Purpose?

Data Abstraction is the property by virtue of which only the essential details are displayed to the user. The trivial or the non-essential units are not displayed to the user. Ex: A car is viewed as a car rather than its individual components. Data Abstraction may also be defined as the process of identifying only the required characteristics of an object ignoring the irrelevant details. The properties and behaviors of an object differentiate it from other objects of similar type and also help in classifying/grouping the objects.

Q.6 Whats Polymorphism and Purpose of it?

Polymorphism is considered one of the important features of Object-Oriented Programming. Polymorphism allows us to perform a single action in different ways. In other words, polymorphism allows you to define one interface and have multiple implementations. The word “poly” means many and “morphs” means forms, So it means many forms.

Q.7 Whats Inheritance and Purpose of it?

Inheritance is an important concept in object oriented programming. In the classical inheritance, methods from base class get copied into derived class.In JavaScript, inheritance is supported by using prototype object. Some people call it "Prototypal Inheritance" and some people call it "Behaviour Delegation".

Q.8 Whats Encapsulation and Purpose of it ?

Encapsulation is the bundling of data and the methods that act on that data such that access to that data is restricted from outside the bundle, or as Alan Kay describes it, “local retention and protection and hiding of state-process.” In OOP, that means that an object stores its state privately, and only the object’s methods have access to change it.

Q.9 Explain Class in JavaScript?

JavaScript classes have introduced as an easy way and a syntactical sugar to write constructor functions. They are mainly used to create new objects.To define a class in JavaScript, we use the keyword `class` and we give it a name with the first letter capitalized. Then we will need to define a constructor method inside of the class.

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

Super keyword in JavaScript can be used to access and call on an object’s parent, it can be used in two ways.

1. As a function
2. As an object