

ANSWER 9–

In JavaScript, a class is a blueprint or a template for creating objects with similar properties and behavior. It is a way to define a reusable structure that encapsulates data (properties) and behavior (methods) within objects. Classes provide a foundation for object-oriented programming in JavaScript.

Here's an explanation of classes in JavaScript:

Class Declaration: A class is declared using the class keyword, followed by the name of the class. It serves as a template for creating objects of that class

```
class Person {  
  // Class definition  
}
```

Constructor: A constructor method is a special method inside a class that is used for initializing objects of the class. It is called automatically when a new object is created using the new keyword.

```
class Person {  
  constructor(name, age) {  
    this.name = name;  
    this.age = age;  
  }  
}
```

In this example, the Person class has a constructor that takes name and age as parameters. Inside the constructor, properties name and age are assigned to the object using this.

Methods: Methods are functions defined within a class that define the behavior of the objects created from the class.

```
class Person {  
  constructor(name, age) {  
    this.name = name;  
    this.age = age;  
  }  
  
  sayHello() {  
    console.log(`Hello, my name is ${this.name} and I am ${this.age} years old.`);  
  }  
}
```

In this example, the Person class has a method sayHello() that logs a greeting with the object's name and age.

Creating Objects: Objects are created from a class using the new keyword followed by the class name and any necessary arguments for the constructor.

```
const john = new Person('John', 25);
```

In this example, the john object is created from the Person class, and the constructor is called with the arguments 'John' and 25.

Classes in JavaScript provide a structured way to define and create objects with shared properties and behavior. They encapsulate data and methods, allowing for code reusability and organization. Classes provide a foundation for object-oriented programming, enabling the creation of complex systems by modeling real-world entities or concepts as objects.