JS (OOP)

Summary Sheet

Qs1. What is Object Oriented Programming (OOP)?

Ans. Object-Oriented Programming (OOP) is a programming paradigm in computer science that relies on the concept of classes and objects. It is used to structure a software program into simple, reusable pieces of code blueprints (usually called classes), which are used to create individual instances of objects.

Qs2. What are some benefits of using OOP in JavaScript?

Ans. Some benefits of using OOP in JavaScript includes:

a. Improved code organization (structure of code)
b. Reusability of code
c. Better maintainability of code
d. Closeness to real-world objects

Qs3. What is the difference between an object and a class in JavaScript?

Ans. Objects in JS is a standalone entity, with properties, methods and a type. It can be created directly from functions or through constructor functions.
Class in JS acts as a blueprint for creating objects.

Qs4. What is a constructor function in JS?

Ans. constructor function is a special function that is used to create & initialize objects in JS. When a new object is created using a constructor function, it is automatically assigned a set of properties and methods that are defined within the function.

Qs5. What is a prototype chain in JavaScript?

Ans. Every object in JavaScript has a built-in property, which is called its prototype. The prototype is itself an object, so the prototype will have its own prototype, making

what's called a prototype chain. The chain ends when we reach a prototype that has null for its own prototype.

Qs6. What is the difference between a constructor and a class in JavaScript?
Ans. A constructor is a function that creates an object, while a class is a blueprint for creating objects. Classes define the framework whereas, constructor actually creates the objects & initializes them.
(In JavaScript, classes are syntactic sugar over constructor functions.)

Qs7. Why is the "new" keyword used in JavaScript?
Ans. The 'new' keyword is used to create an instance of an object. When used with a constructor function, it creates a new object and sets the constructor function's 'this' keyword to point to the new object.

Qs8. What is Inheritance in OOP?
Ans. Inheritance in OOP is defined as the ability of a class to derive properties and characteristics from another class while having its own properties as well.

Qs9. What is the "super" keyword in JS?
Ans. The super keyword in JavaScript acts as a reference variable to the parent class. It is mainly used when we want to access a variable, method, or constructor in the base class from the derived class.

Qs10. What will be the output for the following code:

```
class Box {
  constructor(name, l, b) {
    this.name = name;
    this.l = l;
    this.b = b;
  }

  area() {
    let area = this.l * this.b;
    console.log(`Box area is ${area}`);
  }
}

class Square extends Box {
  constructor(a) {
    super("square", a, a);
  }

  area() {
    let area = this.l * this.b;
    console.log(`Square area is ${area}`);
  }
}

let sq1 = new Square(4);
sq1.area();
```

Ans. The output will be “Square area is 16” as the child class (Square) implementation of area() function will override parent class (Box) implementation of the function with the same name.