

CLases & Objects :-

1. Class:

```
class Car {  
  constructor(make, model) {  
    this.make = make;  
    this.model = model;  
  }  
  
  displayInfo = () => {  
    console.log(`Car: ${this.make} ${this.model}`);  
  }  
}  
  
const myCar = new Car("Toyota", "Corolla");  
myCar.displayInfo(); // Output: Car: Toyota Corolla
```

2. Object:

```
const person = {  
  name: "John",  
  age: 30,  
  greet: function () {  
    console.log(`Hello, my name is ${this.name} and I'm ${this.age} years old.`);  
  }  
};  
  
person.greet(); // Output: Hello, my name is undefined and I'm undefined years old.
```

3. Encapsulation:

```
class Counter {  
  constructor() {  
    let count = 0; // Private member  
  
    this.increment = () => {  
      count++;  
      console.log(count);  
    };  
  
    this.decrement = () => {
```

```
        count--;  
        console.log(count);  
    };  
}  
}  
  
const counter1 = new Counter();  
counter1.increment(); // Output: 1  
counter1.increment(); // Output: 2  
counter1.decrement(); // Output: 1
```

4. Inheritance:

```
class Animal {  
    constructor(name) {  
        this.name = name;  
    }  
  
    speak = () => {  
        console.log(`${this.name} makes a sound.`);  
    }  
}  
  
class Dog extends Animal {  
    speak = () => {  
        console.log(`${this.name} barks.`);  
    }  
}  
  
const dog = new Dog("Buddy");  
dog.speak(); // Output: Buddy barks.
```

5. Polymorphism:

```
// PNB (Punjab National Bank)  
class PNB {  
    rateOfInterest() {  
        return 0.04; // 4% interest rate for PNB  
    }  
}  
  
// SBI (State Bank of India)
```

```

class SBI {
  rateOfInterest() {
    return 0.05; // 5% interest rate for SBI
  }
}

// HDFC (Housing Development Finance Corporation)
class HDFC {
  rateOfInterest() {
    return 0.06; // 6% interest rate for HDFC
  }
}

// Function that accepts any bank and calculates interest
function calculateInterest(bank) {
  const interestRate = bank.rateOfInterest();
  console.log(`Interest Rate for ${bank.constructor.name}: ${interestRate * 100}%`);
}

// Creating instances of different banks
const pnbBank = new PNB();
const sbiBank = new SBI();
const hdfcBank = new HDFC();

// Calling the function with different banks
calculateInterest(pnbBank); // Output: Interest Rate for PNB: 4%
calculateInterest(sbiBank); // Output: Interest Rate for SBI: 5%
calculateInterest(hdfcBank); // Output: Interest Rate for HDFC: 6%

```

6. Abstraction:

```

// Abstract class representing a Shape
class Shape {
  calculateArea() {
    // Default implementation returns 0
    return 0;
  }
}

// Concrete subclass representing a Rectangle
class Rectangle extends Shape {
  constructor(width, height) {
    super();
  }
}

```

```
    this.width = width;
    this.height = height;
  }
  calculateArea() {
    return this.width * this.height;
  }
}
```

```
// Concrete subclass representing a Circle
class Circle extends Shape {
  constructor(radius) {
    super();
    this.radius = radius;
  }
  calculateArea() {
    return Math.PI * this.radius ** 2;
  }
}
```

```
// Creating instances of shapes and calculating their areas
const rectangle = new Rectangle(5, 3);
const circle = new Circle(4);
```

```
console.log("Area of rectangle:", rectangle.calculateArea()); // Output: 15
console.log("Area of circle:", circle.calculateArea().toFixed(2)); // Output: 50.27
```

For More Info Visit :- [Web Dev Mastery](#)