

here are two examples demonstrating various aspects of classes and object-oriented programming concepts in TypeScript:

### ### Example 1: Basic Class Syntax, Constructor, Properties, and Methods

```
``typescript
// Define a class representing a Car
class Car {
  // Properties
  brand: string;
  model: string;
  year: number;

  // Constructor
  constructor(brand: string, model: string, year: number) {
    this.brand = brand;
    this.model = model;
    this.year = year;
  }

  // Method to display car details
  displayDetails(): void {
    console.log(`Brand: ${this.brand}, Model: ${this.model}, Year: ${this.year}`);
  }
}

// Create an instance of Car
let myCar = new Car("Toyota", "Camry", 2020);

// Access properties and call method
myCar.displayDetails();
``
```

### ### Example 2: Inheritance and Method Overriding

```
``typescript
// Define a superclass representing a Shape
class Shape {
  // Properties
  color: string;

  // Constructor
  constructor(color: string) {
    this.color = color;
  }
}
```

```

    }

    // Method to display shape color
    displayColor(): void {
        console.log(`Color: ${this.color}`);
    }
}

// Define a subclass representing a Circle, inheriting from Shape
class Circle extends Shape {
    // Properties
    radius: number;

    // Constructor
    constructor(color: string, radius: number) {
        super(color);
        this.radius = radius;
    }

    // Method to calculate and display area
    displayArea(): void {
        const area = Math.PI * this.radius * this.radius;
        console.log(`Area: ${area.toFixed(2)}`);
    }

    // Method Overriding: Override displayColor method
    displayColor(): void {
        console.log(`Circle Color: ${this.color}`);
    }
}

// Create an instance of Circle
let myCircle = new Circle("Red", 5);

// Access properties and call methods
myCircle.displayColor(); // Output: Circle Color: Red
myCircle.displayArea(); // Output: Area: 78.54
```

```

These examples showcase the basic syntax of classes in TypeScript, including constructors, properties, methods, inheritance, and method overriding.