

1. Introduction
  2. What is a class?
  3. What is an object?
  4. Differences between C++/Java classes and TypeScript classes
- 

## 1. Basic Class and Object

```
class Person {  
    public firstName: string;  
    public lastName: string;  
  
    constructor(firstName: string, lastName: string) {  
        this.firstName = firstName;  
        this.lastName = lastName;  
    }  
  
    fullName(): string {  
        return this.firstName + " " + this.lastName;  
    }  
}  
  
const person1 = new Person("John", "Doe");  
console.log(person1.fullName()); // John Doe
```

## 2. Explain: constructor, public properties, methods, object creation

---

### 1. Private and Protected Members

```
class BankAccount {  
    private balance: number;  
  
    constructor(initialBalance: number) {  
        this.balance = initialBalance;  
    }  
  
    deposit(amount: number) {  
        this.balance += amount;  
    }  
  
    getBalance(): number {
```

```

        return this.balance;
    }
}

const account = new BankAccount(100);
account.deposit(50);
console.log(account.getBalance()); // 150

```

---

2. Explain private properties and accessors

1. Inheritance

```

class Employee extends Person {
    public jobTitle: string;

    constructor(firstName: string, lastName: string, jobTitle: string) {
        super(firstName, lastName);
        this.jobTitle = jobTitle;
    }

    info(): string {
        return `${this.fullName()} works as ${this.jobTitle}`;
    }
}

const emp = new Employee("Alice", "Smith", "Developer");
console.log(emp.info()); // Alice Smith works as Developer

```

---

2. Explain extends, super(), method overriding

1. Static Members

```

class Calculator {
    static pi: number = 3.14159;

    static circleArea(radius: number): number {
        return Calculator.pi * radius * radius;
    }
}

console.log(Calculator.circleArea(5)); // 78.53975

```

---

2. Explain static properties and methods

---

## 1. Getters and Setters

```
class Rectangle {  
    private _width: number;  
    private _height: number;  
  
    constructor(width: number, height: number) {  
        this._width = width;  
        this._height = height;  
    }  
  
    get area(): number {  
        return this._width * this._height;  
    }  
  
    set width(value: number) {  
        if (value > 0) this._width = value;  
    }  
}  
  
const rect = new Rectangle(10, 5);  
console.log(rect.area); // 50  
rect.width = 20;  
console.log(rect.area); // 100
```

---

## 2. Explain get and set keywords

- 
1. Practice Problems
  2. Create a Car class with make, model, year. Add a method age() that returns car age.
  3. Create a Student class with private grades array. Add methods to addGrade() and getAverage().
  4. Create a Shape base class and extend it with Circle and Square classes. Add methods to calculate area.

- 
1. Summary
  2. Classes are blueprints for objects.
  3. Use public, private, protected for encapsulation.
  4. Use extends for inheritance.
  5. Static members belong to the class itself.
  6. Getters and setters allow controlled access to private data.