```
// Task 1: Arrow function to calculate square
const square = num => num * num;
console.log("Square of 5:", square(5));

// Task 2: Array operations on ages
const ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24];
ages.sort((a, b) => a - b); // Sorting array
const minAge = ages[0];
const maxAge = ages[ages.length - 1];

// Finding Median
const mid = Math.floor(ages.length / 2);
const medianAge = ages.length % 2 !== 0 ? ages[mid] : (ages[mid - 1] + ages[mid]) / 2;

// Finding Average
const averageAge = ages.reduce((sum, age) => sum + age, 0) / ages.length;

// Finding Range
const ageRange = maxAge - minAge;

// Comparing min - avg and max - avg
const diffMinAvg = Math.abs(minAge - averageAge);
const diffMaxAvg = Math.abs(maxAge - averageAge);

console.log({ minAge, maxAge, medianAge, averageAge, ageRange, diffMinAvg, diffMaxAvg });

// Task 3: Contact Map
const contacts = new Map();
contacts.set("Alice", { age: 25, email: "alice@example.com", location: "NY" });
contacts.set("Bob", { age: 30, email: "bob@example.com", location: "LA" });

const getContactDetails = name => contacts.get(name) || "Contact not found";
console.log(getContactDetails("Alice"));

// Task 4: Using call method
const person1 = { name: "John", age: 28 };
const person2 = { name: "Jane", age: 32 };

function introduce() {
    console.log(`Hello, I'm ${this.name}, and I'm ${this.age} years old.`);
}

introduce.call(person2); // Using call
```

```
// Task 5: Using Set and Map for unique numbers and their squares
const numbers = new Set([1, 2, 3, 4, 5]);
const squaresMap = new Map();
numbers.forEach(num => squaresMap.set(num, square(num)));

console.log("Numbers:", [...numbers]);
console.log("Squares:", [...squaresMap]);

// Task 6: Using call, apply, and bind
function displayInfo(name, role) {
    console.log(`${name} is a ${role}.`);
}
displayInfo.call(null, "Alice", "Developer"); // Using call
displayInfo.apply(null, ["Bob", "Designer"]); // Using apply

function greet() {
    console.log(`Hello, my name is ${this.name}.`);
}
const boundGreet = greet.bind({ name: "Charlie" });
boundGreet(); // Using bind

// Task 7: Calculator Object with dynamic operations
const calculator = {
    add: (a, b) => a + b,
    subtract: (a, b) => a - b,
    multiply: (a, b) => a * b,

    calculate: function(operation, a, b) {
        return this[operation](a, b);
    }
};

// Using call and apply with calculator
console.log("Addition using call:", calculator.calculate.call(calculator, "add", 5, 3));
console.log("Multiplication using apply:", calculator.calculate.apply(calculator, ["multiply", 4, 2]));

// Task 7: Discount Calculator with bind
const discountCalculator = {
    discount: 10,
    applyDiscount: function(amount) {
        return amount - (amount * this.discount / 100);
    }
};
```

```
const calculateDiscount = discountCalculator.applyDiscount.bind(discountCalculator);
console.log("Discounted Price:", calculateDiscount(100));
```