

1. You have to take 4 inputs with the help of arrow function and calculate its sum.

Ans.

```
const sumNumbers = (a, b, c, d) => a + b + c + d;

const takeInputs = () => {
  let inputs = [];
  for (let i = 1; i <= 4; i++) {
    let input = prompt('\t');
    inputs.push(Number(input)); }
  return inputs;
}

const inputs = takeInputs();
const sum = sumNumbers(...inputs);

console.log(`The sum of the four numbers is: ${sum}`);
```

2. using anonymous arrow function create a function which does factorial of 5.

Ans.

```
const factorial = (() => {
  const num = 5;
  let fact = 1;
  for (let i = 1; i <= num; i++) {
    fact *= i;
  }
  return fact;
})();

console.log(`Factorial of 5 is: ${factorial}`);
```

3. use the concept of hoisting and call a factorial function without declaring it and then define after calling.

Ans.

```
const result = factorial(5);
console.log(`Factorial of 5 is: ${result}`);

function factorial(n) {
```

```

if (n === 0 || n === 1) {
    return 1;
} else {
    return n * factorial(n - 1); }
}

```

4. Use the concept of immediately invoked function to check whether a number is even or odd.

Ans.

```

var userInput = prompt("enter number");

```

```

var EvenOdd = (function(num) {
    if (num % 2 === 0) {
        return `${num} is even.`;
    } else {
        return `${num} is odd.`; }
})(userInput);
console.log(EvenOdd);

```

5. using the concept of parameterised function constructor call factorial function

```

function Factorial(number) {
    this.number = number;

    this.calculate = function() {
        if (this.number < 0) {
            return "Factorial is not defined for negative numbers.";
        }
        if (this.number === 0 || this.number === 1) {
            return 1; }
        let result = 1;
        for (let i = 2; i <= this.number; i++) {
            result *= i; }
        return result;
    };
}

```

```
}
```

```
let fact = new Factorial(7);
```

```
console.log(fact.calculate());
```