app.js > ...

```js
const { add, subtract } = require('./math');

console.log('=== Math Operations Demo ===');

const num1 = 10;
const num2 = 5;

const sum = add(num1, num2);
console.log(`${num1} + ${num2} = ${sum}`);

const difference = subtract(num1, num2);
console.log(`${num1} - ${num2} = ${difference}`);

console.log('Additional examples:');
console.log(`20 + 15 = ${add(20, 15)}`);
console.log(`50 - 25 = ${subtract(50, 25)}`);
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   ···   Filter   Code

```
20 + 15 = 35
50 - 25 = 25

[Done] exited with code=0 in 0.183 seconds

[Running] node "d:\p 3 node js\app.js"
=== Math Operations Demo ===
10 + 5 = 15
10 - 5 = 5
Additional examples:
20 + 15 = 35
50 - 25 = 25

[Done] exited with code=0 in 0.18 seconds
```

math.js > ...

```js
function add(a, b) { ...
}

function subtract(a, b) {
    return a - b;
}

module.exports = {
    add,
    subtract
};
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   ···   Filter   Code

```
20 + 15 = 35
50 - 25 = 25

[Done] exited with code=0 in 0.183 seconds

[Running] node "d:\p 3 node js\app.js"
=== Math Operations Demo ===
10 + 5 = 15
10 - 5 = 5
Additional examples:
20 + 15 = 35
50 - 25 = 25

[Done] exited with code=0 in 0.18 seconds
```

non-blocking-read.js > ...

```js
fs.readFile('sample.txt', 'utf8', (error, data) => {
    if (error) {
        console.error('Error reading file:', error.message);
        return;
    }

    console.log('4. File read completed! (This runs when file is ready)');
    console.log('5. File contents:');
    console.log('-------------------');
    console.log(data);
    console.log('-------------------');
    console.log('6. This callback executed asynchronously!');
});

console.log('3. This runs IMMEDIATELY after starting file read');
console.log('7. Script main execution completed');
console.log('8. Notice: The file contents appear AFTER this message');
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   ···   Filter   Code

```
=== NON-BLOCKING FILE READ DEMO ===
1. Starting script execution...
2. About to read file (NON-BLOCKING)...
3. This runs IMMEDIATELY after starting file read
7. Script main execution completed
8. Notice: The file contents appear AFTER this message
4. File read completed! (This runs when file is ready)
5. File contents:
-------------------
Student Information System
==========================

Name: Akshat Singh
Student ID: GF202344124
Course: Computer Science Engineering
Semester: 5th Semester
```

Welcome to Copilot

Let's get started

Add context (#), extensions (@), commands

Build Workspace   Show Config

Review AI output carefully before use.

blocking-read.js

```js
const fs = require('fs');

console.log('=== BLOCKING FILE READ DEMO ===');
console.log('1. Starting script execution...');

try {
    console.log('2. About to read file (BLOCKING)...');

    const data = fs.readFileSync('sample.txt', 'utf8');

    console.log('3. File read completed!');
    console.log('4. File contents:');
    console.log('-------------------');
    console.log(data);
    console.log('-------------------');

} catch (error) {
    console.error('Error reading file:', error.message);
}
```

OUTPUT

```
=== BLOCKING FILE READ DEMO ===
1. Starting script execution...
2. About to read file (BLOCKING)...
3. File read completed!
4. File contents:
-------------------
Student Information System
==========================

Name: Akshat Singh
Student ID: GF202344124
Course: Computer Science Engineering
Semester: 5th Semester
Email: akshat.singh@university.edu

Current Subjects:
```

---

settimeout.js

```js
function fetchUserData() {
    console.log('Fetching user data...');

    setTimeout(() => {
        console.log('Data received');


        const userData = {
            name: 'Akshat',
            email: 'AKshat@example.com'
        };

        console.log('User:', userData);
    }, 2000);
}


    fetchUserData();
```

OUTPUT

```
-------------------
5. Script execution completed.
6. Notice: All console.log statements execute in order

[Done] exited with code=0 in 0.159 seconds

[Running] node "d:\p 3 node js\settimeout.js"
Fetching user data...
Data received
User: { name: 'Akshat', email: 'AKshat@example.com' }

[Done] exited with code=0 in 2.186 seconds
```