Assignment 1 – Function & Scope

Create a function `calculateTax(income)`:

- If income > 10,00,000 → 30%
- If income > 5,00,000 → 20%
- Else → 10%

Requirements:

- Use arrow function
- Print final tax amount
- Try both `var` and `let` inside function and observe scope behaviour

---

Assignment 2 – Hoisting Experiment

1. Call a function before declaring it (function declaration).
2. Call a function expression before declaring it.
3. Try accessing a `var`, `let`, and `const` variable before initialization.

Assignment 3 – Array Methods (map, filter, reduce)

Given:

```
const employees = [
  { name: "Aarjav", salary: 40000 },
  { name: "Aditya", salary: 60000 },
  { name: "Shivank", salary: 80000 },
  { name: "Riya", salary: 30000 }
];
```

Tasks:

1. Get employees with salary > 50000.
2. Increase all salaries by 10%.
3. Calculate total salary payout.
4. Print only employee names.

Assignment 4 – Object & Class

Create a class `BankAccount`:

Properties:

- accountHolder
- balance

Methods:

- deposit(amount)
- withdraw(amount)
- checkBalance()

Create 2 accounts and perform transactions.

Assignment 5 – DOM Manipulation Mini Project

Create a simple TODO app:

- Input field
- Add button
- Display list items
- Remove item on click

Use:

- `querySelector`
- `addEventListener`
- `createElement`
- `appendChild`

Bonus:

- Change background color when task completed

Assignment 6 – Event Handling

Create:

- A button
- On click → Change page background color randomly
- Show click count

Bonus:

- Reset button to reset count

Assignment 7 – Promise Creation

Create a Promise:

- If random number > 5 → resolve("Success")
- Else → reject("Failed")

Handle using:

- `.then()`
- `.catch()`

Assignment 8 – Convert Promise to async/await

Given Promise-based function:

```
function getData() {
  return new Promise(resolve => {
    setTimeout(() => resolve("Data Received"), 2000);
  });
}
```

Tasks:

1. Call using `.then()`
2. Convert using `async/await`
3. Add try-catch

Assignment 9 – Fetch API

Use this API:

`https://jsonplaceholder.typicode.com/users`

Tasks:

1. Fetch data
2. Print only names
3. Display names inside HTML list dynamically
4. Handle errors

- Show loading message before data loads

Assignment 10 – ES6+ Features Combined

Given:

```
const user = {
  id: 1,
  name: "Jaynam",
  skills: ["JavaScript", "React", "Node"]
};
```

Tasks:

1. Use destructuring to extract name.
2. Use template literals to print:
   "Jaynam knows JavaScript, React, Node"
3. Use spread operator to add new skill.
4. Create function using rest operator to accept multiple skills.