Part A - Theory (Set 1)

1. What is the difference between var, let, and const? Explain with one example each.

- **var**: Function-scoped. Can be re-declared and updated.
    - var name = "John"; var name = "Doe"; // This is allowed
- **let**: Block-scoped ({ }). Can be updated but not re-declared in the same scope.
    - let age = 25; age = 26; // This is allowed
- **const**: Block-scoped. Must be initialized during declaration and cannot be re-assigned.
    - const pi = 3.14; pi = 3.15; // ❌ Error: Assignment to constant variable.

2. List all primitive data types in JavaScript. Give one real-world use case for any two.

Primitive data types: string, number, bigint, boolean, undefined, null, symbol.

- string: Storing a user's name. let userName = "Alice";
- boolean: Toggling a feature on/off. let isLoggedIn = true;

3. What is the difference between == and ===? Which one is better to use in projects?

- == (Loose Equality): Checks only value after converting types if necessary. 5 == '5' is true.
- === (Strict Equality): Checks both value AND type. 5 === '5' is false.
- === **is better** because it avoids unexpected bugs caused by automatic type conversion.

4. Explain with an example how the operator += works in JavaScript.

The += operator is an addition assignment operator. It adds the right operand to the left operand and assigns the result back to the left operand.

- **Example:** let count = 5; count += 3; // count is now 8

5. What is the difference between null and undefined in JavaScript?

- undefined: A variable that has been declared but not assigned a value. It's automatic.
    - let x; console.log(x); // undefined
- null: An assignment value. It represents the intentional absence of any object value.
    - let user = null; // Developer explicitly set this to mean "no user"

Part A - Theory (Set 2)

1. Difference between var, let, const with examples?

- **var**: var score = 5; var score = 10; (Allowed)
- **let**: let score = 5; score = 10; (Allowed). let score = 15; (Error)
- **const**: const score = 5; score = 10; (Error)

2. What is declaration and initialization?

- **Declaration**: Saying a variable exists. let age;
- **Initialization**: Giving it a value for the first time. age = 30;

3. Can we re-assign let and const?

- **let**: Yes. let x = 1; x = 2; is okay.
- **const**: No. const x = 1; x = 2; will cause an error.

4. 4 categories of operators with examples?

1. **Arithmetic**: for math (+, -)
2. **Assignment**: to give values (=, +=)
3. **Comparison**: to compare values (==, >)
4. **Logical**: to combine checks (&& which means AND, || which means OR)

5. Why prefer let and const over var?

- let and const are safer. They are limited to the {} they are in, so they don't cause unexpected errors elsewhere in the code.

Part B - Programming

1. Declare with let, initialize, re-assign, and print.

javascript

```
let number = 10; // Initialize
console.log(number); // Prints 10
number = 20; // Re-assign
console.log(number); // Prints 20
```

2. Use const for PI and try to re-assign.

javascript

```
const PI = 3.14159;
PI = 3; // This line will cause an ERROR in the console.
```

3. Check if a number is greater than 50.

javascript

```
let myNumber = 75;
console.log(myNumber > 50); // This will print 'true'
```

4. Add two numbers and store the result.

javascript

```
let num1 = 5;
let num2 = 3;
let result = num1 + num2; // + is arithmetic, = is assignment
console.log(result); // Prints 8
```

5. Check vote eligibility (age >= 18 AND isCitizen).

javascript

```
let age = 20;

let isCitizen = true;

console.log(age >= 18 && isCitizen); // Prints 'true'
```

Part C - Scenario-Based

1. Shopping cart total bill: var, let, or const? Why?

- **let**. Because the total changes as you add or remove items from the cart.

2. A variable is declared but has no value. What is its value?

- It is automatically undefined.

3. Student's marks (change) vs. roll number (never changes). Which keywords?

- **Marks**: let (because it changes).
- **Roll Number**: const (because it never changes).

4. In a banking app, balance reduces on withdrawal. Which operator?

- **Assignment operators** like -= (e.g., balance -= 100;).

5. Login check: user logged in AND password correct. Which operator?

- A **Logical operator**, specifically the && (AND) operator.