SECTION A: Technical Questions

1. Difference between var, let, and const:

var:

- Function scoped
- Allows redeclaration
- Allows reinitialization

let:

- Block scoped
- Does not allow redeclaration
- Allows reinitialization

const:

- Block scoped
- Does not allow redeclaration
- Does not allow reinitialization

2. Which keyword allows redeclaration and why?

Answer:

var allows redeclaration because it is function scoped and JavaScript does not restrict redeclaring variables declared using var.

3. Which keyword allows reinitialization?

Answer:

var and let allow reinitialization.

4. Which keyword does not allow redeclaration and reinitialization?

Answer:

const does not allow redeclaration and reinitialization.

5. Why should const be used for fixed values?

Answer:

const should be used for fixed values because it prevents accidental changes and makes the code safer and more readable.

6. What error occurs when redeclaring a let variable?

SyntaxError occurs because let does not allow redeclaration in the same scope.

7. What error occurs when reassigning a const variable?

Answer:

TypeError occurs because const variables cannot be reassigned.

8. Which keyword is preferred in modern JavaScript and why?

Answer:

let and const are preferred because they provide block scope and reduce bugs compared to var.

9. Can const be declared without initialization? Explain.

Answer:

No, const cannot be declared without initialization because a value must be assigned at the time of declaration.

10. When should var be avoided?

Answer:

var should be avoided because it is function scoped, allows redeclaration, and can cause unexpected bugs.

SECTION B: CODE-BASED QUESTIONS

11. Predict the output:

var a = 10;

a = 20;

var a = 30;

console.log(a);

Answer:

30

Reason:

var allows redeclaration and reinitialization.

12. Predict the output:

let b = 5;

b = 15;

console.log(b);

Answer:

15

13. Identify the error:

let x = 10;

let x = 20;

Answer:

SyntaxError

Reason:

Redeclaration of let variable is not allowed.

14. Identify the error:

const y = 50;

y = 100;

Answer:

TypeError

Reason:

const variables cannot be reassigned.

15. Program using var to show redeclaration:

```
var name = "Sai";
var name = "Venkatesh";
console.log(name);
```

output:venkatesh

16. Program using let to show reinitialization:

```
let age = 21;
age = 22;
console.log(age);
```

output:22

17. Program using const and explanation:

```
const PI = 3.14;
PI = 3.14159
console.log(PI);
```

Explanation:

const value cannot be changed after initialization, so it throws TypeError.

18. Convert var to let:

```
var count = 10;
let count = 10;
console.log(count);
```

19. Convert var to const:

```
var PI = 3.14;
const PI = 3.14;
console.log(PI);
```

20. Own example for var, let, and const:

```
var city = "Hyderabad";
let marks = 90;
const college = "JNTU";
```