DAY-9 ASSIGNMENT (Swami Vivekananda University)

22 Feb 2025

◆ Section 1: Data Types & Length

1 **What will be the output of the following code? Explain why.**

```
let x = "5";

let y = 5;

console.log(x == y);

console.log(x === y);
```

2 **How can you find the length of the longest word in this array?**

```
const words = ["JavaScript", "Programming", "Function", "Hoisting"];

// Write a function to get the longest word length
```

◆ Section 2: let, const, var & Scope

3 **What will be logged in the console? Explain your answer.**

```
function testScope() {

   if (true) {

      var a = 10;

      let b = 20;

      const c = 30;

   }

   console.log(a);

   console.log(b);

   console.log(c);

}

testScope();
```

4 **Rewrite the following function using const and fix any issues.**

```
var name = "John";

function greet() {

   if (true) {
        var message = "Hello " + name;
    }
    console.log(message);
}
greet();
```

Section 3: Traditional Function vs. Arrow Function

5 **Convert this traditional function into an arrow function without changing the output.**

```
function multiply(a, b) {
    return a * b;
}
```

6 **What will be the output? Explain why.**

```
const obj = {
    name: "Alice",
    sayHello: function() {
        setTimeout(() => {
            console.log("Hello, " + this.name);
        }, 1000);
    }
};
obj.sayHello();
```

Section 4: Hoisting

7 **What will be the output of this code? Explain why.**

```
console.log(a);
var a = 10;
console.log(b);
let b = 20;
```

8 **Rearrange the following function so it works correctly, explaining the problem.**

```
console.log(square(5));
```

```
function square(n) {
  return n * n;
}
console.log(double(4));

var double = function(n) {
  return n * 2;
};
```

◆ Section 5: Operators (Arithmetic, Assignment, Comparison, Logical)

9 **What will be the output of the following expressions?**

```
console.log(5 + "5");
console.log(5 - "3");
console.log(5 * "2");
console.log("10" / 2);
console.log(10 % "3");
```

10 **Rewrite this code using shorthand assignment operators.**

```
let x = 10;
x = x + 5;
x = x * 2;
x = x - 3;
x = x / 2;
```

11 **What will be logged in the console? Explain why.**

```
console.log(5 > 3 && 10 < 20);
console.log(5 > 3 || 10 > 20);
console.log(!(5 > 3));
```

◆ Section 6: Function Parameters vs Arguments

12 **What will be the output of the following function?**

```
function sum(a, b, c = 5) {
  return a + b + c;
}
console.log(sum(2, 3));
console.log(sum(2, 3, 10));
```

13 **Write a function that takes any number of arguments and returns their sum.**

🔹 **Bonus: Hard Challenge Question**

14 **Fix the following function to work correctly and explain the issues.**

```
function outer() {
  let count = 0;
  return function inner() {
    count++;
    console.log(count);
  };
}
const counter1 = outer();
counter1();
counter1();
const counter2 = outer();
counter2();
counter2();
```

##Some statement questions:

1 Number Reversal without Using Built-in Methods

👉 **Problem:**
Write a function reverseNumber(num) that takes a number and returns its reverse.
Example:

```
reverseNumber(1234); // Output: 4321
reverseNumber(-567); // Output: -765
```

⚡ **Hint:** Convert the number to a string manually and reverse it.

---

2 Custom Length Function

👉 **Problem:**
Create a function customLength(str) that returns the length of a string **without using length property**.
Example:

```
customLength("JavaScript"); // Output: 10
```

3 Avoid Hoisting Bug

👉 **Problem:**
Fix the bug in the following code **without changing the function calls**.

```
console.log(add(2, 3)); // Should output: 5

console.log(multiply(2, 3)); // Should output: 6


function add(a, b) {

   return a + b;

}


var multiply = function(a, b) {

   return a * b;

};
```

⚡ **Hint:** Understand how function declarations and function expressions behave with hoisting.

4 Function That Returns Another Function

👉 **Problem:**
Write a function counter() that returns another function. Each time the returned function is called, it should **increase a count** and return it.
Example:

```
const count = counter();

console.log(count()); // Output: 1

console.log(count()); // Output: 2

console.log(count()); // Output: 3
```

⚡ **Hint:** Use closures.