Conceptual Questions

1. Why are promises used in JavaScript? Explain the advantages of using promises over traditional callback functions.

Promise is an object designed to deliver a single value at some point in the future. It is used in JavaScript due to its ability to write cleaner code but reduce (or entirely remove) callback hell. It provides a way to handle asynchronous operations by representing the eventual completion or failure. It has three states: pending, fulfilled, or rejected.

Advantages :

Avoid callback hell which can be unreadable.

Makes it easy to write parallel asynchronous code with ‘Promise.all()’.

Generally, more readable and organized.

Error Handling

Support for Async / Await – Allows to write asynchronous code in synchronous style.

2. What is a closure in JavaScript? Provide an example.

Clouser is a feature in JavaScript where an inner function has outer enclosing function’s variable. It has access to variables in its parent scope, even after the parent function has returned.

Clouser are functions that refer to independent (free) variables.

Free variables are variables that are neither locally declared nor passed as parameters.

Example:

```
Function numberGenerator () {
let num = 1;
```

```
function checkNumber() {
        console.log((num);
}
  num++;
  return checkNumber;
}
var number = numberGenerator();
number();
```

Output : When number(); is called, it will log 2 to the console.

3. What is a callback function and why is it used in JavaScript?

Callback function is a function that is passed as an argument of another function. Callbacks are frequently used to manage asynchronous operations like API requests, event handlers and timers.

Callback function is used in JavaScript because they make sure that a function is not going to run before a task is completed but will run right after the task has completed. It helps us develop asynchronous JavaScript code and keep it safe from problems and errors.

In JavaScript, callback function is passed as an argument to another function. The callback is then invoked after certain events occur or a specific task is finished.

4. What are async/await in JavaScript and how do they improve asynchronous

programming?

Async and Await in JavaScript are the keywords used to handle asynchronous operations with promises. It is syntactic sugar built on top of promise.

Async Function simply allows us to write promises- based code as if it were synchronous and it checks that we are not breaking the execution thread.

Await Function is used to wait for the promise. It could be used within the async block only. It waits for the code until the promise returns as result.

5. Write the difference between ES6 and JS.

ES6 stands for ECMAScript6, and it allows to make the code more modern and readable. Using this we can write less and do more.

ES6 is JavaScript, it's just a newer syntax.

ES6 -

ES6 is an advanced version of JavaScript with some advanced features. It is easy to understand.

Standardized scripting language, independent of any specific environment.

Provides a standard for compatibility across different implementations.

Specifies the fundamental characteristics of the scripting language.

JavaScript -

JavaScript implements environment-specific features.

Executed by browsers using ECMAScript guidelines; compatibility with the latest features varies by browser.

Used by web developers for client-side and server-side scripting.

6. What are some of the major features introduced in ES6?

ES6 introduced significant changes to the JavaScript language, making it more powerful and expressive for developers. Some of the key features introduced in ES6 include:

- **Default parameters**

In JavaScript, the default value for function parameter is undefined.

- **Template literals (Template strings)**

Template literals allow embedded expressions, multi-line strings and string interpolation.

- **Tagged Templates**

With tagged templates, we can parse template literals with a function.

- **De-structuring assignment**

The de-structuring assignment allows reading values from an array or properties from an object, into distinct variables.

- **Arrow function expressions**

Arrow function expressions is a new syntax to writing ordinary function expressions

- **Let and const**

The let statement declares a block-scoped local variable

- **Spread and Rest syntaxes (...)**

Rest and spread syntax use ... (three dots) notation.

- **Object.assign() and Object.is()**

  Object.assign() method can be used to clone an object.

- **Classes**

They like functions can be defined as class declarations and class expressions and can be declared using the keyword class followed by the name of the class (say, Person).