**Module-1: Node Introduction**

1. **What is Node.js? Where can you use it?**

* It is used for server-side programming, and primarily deployed for non-blocking, event-driven servers, such as traditional web sites and back-end API services, but was originally designed with real-time, push-based architectures in mind.

1. **Explain callback in Node.js.**

* Callback means function call in other function parameter.
* Ex. function demo1(){

….code

}

function demo2(a){

a();

}

demo2(demo1);

1. **What are the advantages of using promises instead of callbacks?**

* Promises offer several advantages over traditional callbacks, making asynchronous JavaScript code more readable, manageable, and less error-prone. Here are some key advantages of using promises:
* Improved Readability
* Chaining
* Error Handling

1. **What is NPM?**

* "Node Package Manager" – is the default package manager for JavaScript's runtime Node.js.
* NPM consists of two main parts:
* a CLI (command-line interface) tool for publishing and downloading packages, and an [online repository](https://www.npmjs.com/) that hosts JavaScript packages.

1. **What are the modules in Node.js? Explain**

* In Node. js, Modules are the blocks of encapsulated code that communicate with an external application on the basis of their related functionality.

1. **Create TODO app using FS module with yargs (command line arguments).**

* **const yargs = require('yargs');**

**let tasks = [];**

**yargs.command({**

**command: 'add',**

**describe: 'Add a new task',**

**builder: {**

**task: {**

**describe: 'Task description',**

**demandOption: true,**

**type: 'string'**

**}**

**},**

**handler: (argv) => {**

**tasks.push(argv.task);**

**console.log('Task added successfully!');**

**console.log('Tasks:');**

**displayTasks();**

**}**

**});**

**yargs.command({**

**command: 'list',**

**describe: 'List all tasks',**

**handler: () => {**

**console.log('Tasks:');**

**displayTasks();**

**}**

**});**

**yargs.command({**

**command: 'delete',**

**describe: 'Delete a task',**

**builder: {**

**index: {**

**describe: 'Task index to delete',**

**demandOption: true,**

**type: 'number'**

**}**

**},**

**handler: (argv) => {**

**if (argv.index >= 1 && argv.index <= tasks.length) {**

**tasks.splice(argv.index - 1, 1);**

**console.log('Task deleted successfully!');**

**} else {**

**console.log('Invalid task index!');**

**}**

**console.log('Tasks:');**

**displayTasks();**

**}**

**});**

**const displayTasks = () => {**

**tasks.forEach((task, index) => {**

**console.log(`${index + 1}. ${task}`);**

**});**

**};**

**yargs.parse();**