

## VIVA QUESTION-ANSWER C-SKILL

### PRACTICAL-01

**Q):** What is Node.js, and why is it considered an open-source server environment?

**Solution:** Node.js is an open-source runtime environment for executing JavaScript code on the server-side.

**Q):** Who developed Node.js?

**Solution:** Ryan Dahl.

**Q):** How does Node.js handle concurrent connections efficiently?

**Solution:** Node.js uses an event-driven, non-blocking I/O model.

**Q):** Name a prominent feature of Node.js that makes it suitable for real-time applications.

**Solution:** WebSocket support for real-time communication.

**Q):** What is NPM in the context of Node.js?

**Solution:** NPM (Node Package Manager) is a package manager for Node.js that helps manage dependencies.

**Q):** Explain the concept of a "callback" in Node.js.

**Solution:** A callback is a function passed as an argument to another function and is executed when the operation is completed.

**Q):** In which programming language is Node.js primarily written?

**Solution:** Node.js is primarily written in JavaScript.

**Q):** Name a popular web framework built on top of Node.js for creating web applications.

**Solution:** Express.js is a popular web framework for Node.js.

**Q):** What is the primary use of the `require` function in Node.js?

**Solution:** The `require` function is used to include external modules in Node.js.

**Q):** How does Node.js handle concurrency compared to traditional server-side technologies like Apache?

**Solution:** Node.js handles concurrency using a single-threaded, event-driven model, which is more efficient for handling a large number of concurrent connections.

## PRACTICAL-02

**Q):** How do you create a basic Node.js program?

**A:** Create a JavaScript file with a `.js` extension and write your code in it.

**Q):** What is the standard way to display output in Node.js?

**A:** Use the `console.log()` function to print messages to the console.

**Q):** Can you explain how to run a Node.js program from the command line?

**A:** Use the `node` command followed by the filename, e.g., `node yourfile.js`.

**Q):** What is the purpose of the `process` object in Node.js?

**A:** The `process` object provides information and control over the Node.js process.

**Q):** How do you pass command-line arguments to a Node.js program?

**A:** You can access command-line arguments using the `process.argv` array.

**Q):** What is the output of `console.log("Hello, World!")`?

**A:** "Hello, World!" is printed to the console.

**Q):** Can you use variables to store and print messages in Node.js?

**A:** Yes, you can use variables to store and display messages.

**Q):** What is the purpose of the `process.stdout` stream in Node.js?

**A:** `process.stdout` is a writable stream used to write data to the standard output.

**Q):** How can you include comments in a Node.js program?

**A:** Use `//` for single-line comments and `/* */` for multi-line comments.

**Q):** Can you run a Node.js program without a console or terminal?

**A:** No, Node.js programs are typically executed in a console or terminal environment.

### PRACTICAL-03

**Q):** What is the HTTP module in Node.js used for? **A:** Handling HTTP requests and responses for building web servers.

**Q):** How do you create a basic HTTP server in Node.js? **A:** Use `http.createServer()` method from the HTTP module.

**Q):** What is the role of a callback function in the `createServer` method? **A:** Handles incoming requests and responses.

**Q):** How do you listen to a specific port on the server? **A:** Use `server.listen(port)` to specify the listening port.

**Q):** What is the HTTP request object in Node.js, and what information does it contain? **A:** Contains data about incoming requests (URL, method, headers).

**Q):** What is the HTTP response object in Node.js, and what can you do with it? **A:** Used to send responses to clients (set headers, write data, end response).

**Q):** How can you serve static files with Node.js? **A:** Read file content and send it as a response to requests for specific files.

**Q):** What is middleware in Node.js and HTTP requests? **A:** Functions for processing requests before or after handling.

**Q):** What is routing in Node.js web applications? **A:** Determining how to respond based on the URL and HTTP method.

**Q):** How can you handle different HTTP request methods (e.g., GET, POST) in a Node.js web server? **A:** Use conditional statements in the request handler function to differentiate methods.

## PRACTICAL-04

**Q):** What are built-in modules in Node.js? **A:** Built-in modules are pre-built modules that are included with Node.js to provide various functionalities.

**Q):** How do you include a built-in module in a Node.js program? **A:** You use the `require` function to include a built-in module.

**Q):** Can you give an example of a built-in module in Node.js? **A:** The `fs` module is a built-in module used for file system operations.

**Q):** What is the purpose of the `os` module in Node.js? **A:** The `os` module provides information about the computer's operating system.

**Q):** How can you use the `os` module to retrieve information about the CPU architecture? **A:** Use the `os.arch()` method to retrieve the CPU architecture.

**Q):** What is the `path` module in Node.js primarily used for? **A:** The `path` module provides utilities for working with file and directory paths.

**Q):** How can you use the `path` module to join two or more directory or file path segments? **A:** You can use the `path.join()` method to join path segments.

**Q):** What does the `util` module offer in Node.js? **A:** The `util` module provides various utility functions.

**Q):** How can you use the `util` module to format a string with variables? **A:** The `util.format()` method allows you to format strings with variables.

**Q):** What is the `http` module used for in Node.js? **A:** The `http` module is used to create HTTP servers and clients.

## PRACTICAL-05

**Q):** What is a custom module in Node.js? **A:** A custom module is a user-defined module created to encapsulate specific functionality for reuse in Node.js programs.

**Q):** How do you create a custom module in Node.js? **A:** You create a JavaScript file with your module code and use the `module.exports` object to expose functions or variables.

**Q):** What is the purpose of the `require` function when using a custom module? **A:** The `require` function is used to include and access the functionality of a custom module in your Node.js program.

**Q):** Can you give an example of a custom module and how it is used in a Node.js program? **A:** Sure, for instance, if you create a module that calculates the area of a circle, you can use it in your program like this: `const circle = require('./circle');`

**Q):** How can you pass data to a function in a custom module? **A:** You pass data as function parameters when you call the function from your Node.js program.

**Q):** What is the primary advantage of using custom modules in Node.js? **A:** Reusability, as you can encapsulate and share specific functionality across multiple Node.js programs.

**Q):** Is it possible to have multiple functions and variables in a single custom module? **A:** Yes, you can have multiple functions and variables in a single custom module.

**Q):** How do you organize and structure your custom modules in a Node.js project? **A:** You can create a dedicated folder for custom modules and organize them based on their purpose or functionality.

**Q):** Can custom modules depend on other custom modules? **A:** Yes, custom modules can depend on and require other custom modules to build complex functionality.

**Q):** What is the difference between built-in modules and custom modules in Node.js? **A:** Built-in modules are provided by Node.js, while custom modules are created by users for specific purposes.

## PRACTICAL-06

**Q):** What is the role of the file system module in Node.js? **A:** The file system module (`fs`) in Node.js is used to interact with the file system, allowing you to read, write, and manipulate files.

**Q):** How do you include the file system module in a Node.js program? **A:** You include the file system module using `require('fs')`.

**Q):** What is the primary method for reading the contents of a file using the `fs` module? **A:** The `fs.readFile()` method is used to read the contents of a file.

**Q):** How do you write data to a file using the `fs` module? **A:** You can use the `fs.writeFile()` method to write data to a file.

**Q):** What is the difference between synchronous and asynchronous file system methods? **A:** Synchronous methods block the program until the operation is complete, while asynchronous methods allow the program to continue executing other tasks while the operation is in progress.

**Q):** How can you check if a file or directory exists using the `fs` module? **A:** The `fs.existsSync()` method can be used to check the existence of a file or directory.

**Q):** What is the purpose of the `fs.stat()` method in the `fs` module? **A:** The `fs.stat()` method provides information about a file, such as its size and modification time.

**Q):** Can you create directories with the `fs` module? **A:** Yes, you can use the `fs.mkdir()` method to create directories.

**Q):** How do you delete files using the `fs` module? **A:** The `fs.unlink()` method is used to delete files.

**Q):** What is the primary use of the `fs.rename()` method in the `fs` module? **A:** The `fs.rename()` method is used to rename files or directories.

## PRACTICAL-07

**Q):** What is the common use case for handling file requests in a Node.js web server?

**A:** Serving static files (e.g., HTML, CSS, JavaScript) to clients.

**Q):** Which module is typically used for file operations in this context?

**A:** The `fs` module (File System).

**Q):** What is the HTTP status code for "File Not Found" in web development?

**A:** HTTP status code 404.

**Q):** How can you check if a file exists before attempting to read it?

**A:** Use `fs.existsSync()` to check file existence.

**Q):** What method is commonly used to read file content asynchronously?

**A:** `fs.readFile()`.

**Q):** Where should you handle and respond to file requests in a Node.js web server?

**A:** Typically in a request handler function for specific routes or paths.

**Q):** How can you send an HTTP response with a 404 status code in Node.js?

**A:** Use `res.writeHead(404, { 'Content-Type': 'text/plain' });` followed by `res.end('404 Not Found');`.

**Q):** What is the purpose of the "Content-Type" header when sending an HTTP response?

**A:** Specifies the type of content being sent (e.g., "text/html" for HTML files).

**Q):** Why is it important to handle errors and provide meaningful error messages when serving files?

**A:** Enhances user experience and aids in troubleshooting.

**Q):** Can you serve different types of files using the same code for handling file requests in Node.js?

**A:** Yes, by setting the "Content-Type" header based on the file type being requested.

## PRACTICAL-08

**Q):** What is npm in the context of Node.js? **A:** npm is the Node Package Manager used for managing packages and libraries in Node.js.

**Q):** What is the purpose of installing packages using npm? **A:** Installing packages allows access to external libraries and modules for Node.js projects.

**Q):** How do you initiate a Node.js project and create a `package.json` file? **A:** Use `npm init` to create a `package.json` file.

**Q):** What is the 'Upper-case' package in npm used for? **A:** The 'Upper-case' package is used to convert text to uppercase.

**Q):** How do you install the 'Upper-case' package using npm? **A:** Use `npm install upper-case` to install the package.

**Q):** What is the typical workflow for using an npm package in a Node.js program? **A:** Require the package using `const packageName = require('package-name')`.

**Q):** How do you use the 'upper-case' package to convert text to uppercase in your program? **A:** Require the package, then use `upperCase('text')` to convert text to uppercase.

**Q):** Can you specify package dependencies in the `package.json` file? **A:** Yes, dependencies can be specified in the `dependencies` section of the `package.json` file.

**Q):** What is the purpose of using uppercase text conversion in a Node.js program? **A:** It is useful for standardizing text input or formatting text for display.

**Q):** Are there other npm packages available for text manipulation in Node.js? **A:** Yes, there are numerous npm packages available for various text manipulation tasks in addition to text conversion to uppercase.