Q-A)

1. **\*\*What is the command to initialize node package manager (NPM)? Write its syntax.\*\***

   - The command to initialize NPM is `npm init`. This command creates a `package.json` file, which contains metadata about the project and its dependencies. The syntax is:

     ```sh

     npm init

     ```

2. **\*\*What is REPL?\*\***

   - REPL stands for Read-Eval-Print Loop. It is an interactive shell that processes Node.js expressions. It reads the input, evaluates it, prints the result, and loops back to read the next input. REPL is useful for testing and debugging code snippets in real-time.

3. **\*\*List any four core modules of Node.js:\*\***

   - `http`: Used to create HTTP servers and clients.

   - `fs`: Provides an API for interacting with the file system.

   - `path`: Provides utilities for working with file and directory paths.

   - `os`: Provides operating system-related utility methods and properties.

4. **\*\*List any two methods included under the path module of Node.js.\*\***

   - `path.join()`: Joins all given path segments together using the platform-specific separator as a delimiter, then normalizes the resulting path.

   - `path.resolve()`: Resolves a sequence of paths or path segments into an absolute path.

5. **\*\*For which tasks is a File System module used?\*\***

   - Reading files: `fs.readFile()`

   - Writing files: `fs.writeFile()`

   - Deleting files: `fs.unlink()`

   - Renaming files: `fs.rename()`

6. **\*\*Write a command to add dependency “express” using NPM.\*\***

   - The command to add the `express` dependency using NPM is:

     ```sh

     npm install express

     ```

7. **\*\*Write a command to install YSOL Package by using NPM.\*\***

   - The command to install the `ysol` package using NPM is:

     ```sh

     npm install ysol

     ```

8. **\*\*Write down steps to handle HTTP requests while creating a web server using Node.js.\*\***

   - Import the `http` module using `require('http')`.

   - Create a server using `http.createServer()`.

   - Define a callback function to handle requests and responses. This function receives `req` (request) and `res` (response) objects.

   - Listen on a specific port using `server.listen(port)`. This makes the server start listening for incoming requests.

9. **\*\*Write any two advantages of Node.js.\*\***

   - Non-blocking I/O: Node.js uses asynchronous, event-driven architecture, which allows it to handle many connections simultaneously without blocking the execution.

   - Single-threaded event loop: Node.js operates on a single-threaded event loop, which makes it lightweight and efficient for handling concurrent operations.

10. **\*\*Write any two functions of Buffer used in Node.js.\*\***

    - `Buffer.alloc(size)`: Allocates a new buffer of the specified size.

    - `Buffer.from(array)`: Creates a new buffer containing a copy of the provided array of bytes.

11. **\*\*What is the command to initialize node package manager? Write its syntax.\*\***

    - The command to initialize NPM is `npm init`. This command creates a `package.json` file, which contains metadata about the project and its dependencies. The syntax is:

      ```sh

      npm init

      ```

12. **\*\*For which task is a file system module used?\*\***

    - The file system module is used for reading and writing files, among other tasks. For example, `fs.readFile()` reads the content of a file, and `fs.writeFile()` writes data to a file.

13. **\*\*What is express.js?\*\***

    - Express.js is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It simplifies the process of building web servers and APIs by providing a set of middleware functions and routing capabilities.

14. **\*\*What is the use of prompt-sync module?\*\***

    - The `prompt-sync` module is used to get synchronous user input from the command line. It allows you to prompt the user for input and wait for their response before continuing execution.

15. **\*\*List any four core modules in Node.js.\*\***

    - `http`: Used to create HTTP servers and clients.

    - `fs`: Provides an API for interacting with the file system.

    - `path`: Provides utilities for working with file and directory paths.

    - `os`: Provides operating system-related utility methods and properties.

16. **\*\*Which command is used for deleting a file?\*\***

    - The command to delete a file using the `fs` module is:

      ```sh

      fs.unlink(path, callback)

      ```

17. **\*\*Write syntax to create Buffer.\*\***

    - The syntax to create a buffer in Node.js is:

      ```js

      const buffer = Buffer.alloc(size);

      ```

18. **\*\*Write a command to install MySQL package by using NPM.\*\***

    - The command to install the `mysql` package using NPM is:

      ```sh

      npm install mysql

      ```

19. **\*\*What is Node.js?\*\***

    - Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It allows you to run JavaScript on the server side, enabling the development of scalable and high-performance network applications.

20. **\*\*Which type of applications can we build using Node.js?\*\***

    - Web applications: Dynamic websites and web services.

    - Real-time applications: Chat applications, online gaming, and collaboration tools.

    - API services: RESTful APIs and GraphQL APIs.

    - Command-line tools: Utilities and scripts for automation.

21. **\*\*What is the use of Registry?\*\***

    - The registry is a database of JavaScript packages, where developers can publish and share their code. NPM uses the registry to manage and distribute packages.

22. **\*\*Define Anonymous function.\*\***

    - An anonymous function is a function without a name. It is often used as an argument to other functions or as an immediately invoked function expression (IIFE). Anonymous functions are useful for creating inline functions and callbacks.

23. **\*\*Explain Web Server.\*\***

    - A web server is a server that handles HTTP requests and serves web pages to clients. In Node.js, you can create a web server using the `http` module. The server listens for incoming requests, processes them, and sends back the appropriate response.

24. **\*\*What is Package.json file?\*\***

    - `package.json` is a file that contains metadata about a Node.js project, including its dependencies, scripts, and version. It is used by NPM to manage the project's dependencies and scripts.

25. **\*\*Explain global package.\*\***

    - A global package is a Node.js package that is installed globally on your system, making it accessible from any directory. Global packages are typically used for command-line tools and utilities.

26. **\*\*List out the parameters of CreateConnection().\*\***

    - `host`: The hostname of the database server.

    - `user`: The username to connect to the database.

    - `password`: The password to connect to the database.

    - `database`: The name of the database to connect to.

27. **\*\*Write down the Syntax to Concatenate Node buffers to a Single Node Buffer.\*\***

    - The syntax to concatenate multiple Node buffers into a single buffer is:

      ```js

      Buffer.concat([buffer1, buffer2, ...]);

      ```

28. **\*\*Write down the types of modules in Node.js.\*\***

    - Core modules: Built-in modules provided by Node.js, such as `http`, `fs`, and `path`.

    - Local modules: Custom modules created by the developer.

    - Third-party modules: Modules created by the community and published on the NPM registry.

Q-B)

1. **\*\*Explain steps to install Node.JS on windows.\*\***

    - Download the Node.js installer from the official website.

    - Run the installer and follow the prompts.

    - Verify the installation by running `node -v` and `npm -v` in the command prompt.

2. **\*\*Write a Program to write to a file in Node.js\*\***

    ```js

    const fs = require('fs');

    fs.writeFile('example.txt', 'Hello, world!', (err) => {

        if (err) throw err;

        console.log('File has been saved!');

    });

    ```

3. **\*\*How to add dependency into Package.json?\*\***

    - Use the command `npm install <package-name> --save`.

4. **\*\*Write a Program to calculate factorial of given number using function.\*\***

    ```js

    function factorial(n) {

        if (n === 0) return 1;

        return n \* factorial(n - 1);

    }

    console.log(factorial(5)); // Output: 120

    ```

5. **\*\*Explain working of writeHead().\*\***

    - The `writeHead()` method is used to write the HTTP header to the response. It takes the status code and an optional headers object as arguments.

6. **\*\*Explain Inheriting events with a suitable example.\*\***

    ```js

    const EventEmitter = require('events');

    class MyEmitter extends EventEmitter {}

    const myEmitter = new MyEmitter();

    myEmitter.on('event', () => {

        console.log('An event occurred!');

    });

    myEmitter.emit('event');

    ```

7. **\*\*Write a Program to delete table records using Node.js and MySQL database.\*\***

    ```js

    const mysql = require('mysql');

    const connection = mysql.createConnection({

        host: 'localhost',

        user: 'root',

        password: '',

        database: 'test'

    });

    connection.connect();

    connection.query('DELETE FROM users WHERE id = 1', (err, result) => {

        if (err) throw err;

        console.log('Record deleted');

    });

    connection.end();

    ```

8. **\*\*How do you install Packages locally using NPM? Explain with an example.\*\***

    - Use the command `npm install <package-name>`. For example:

      ```sh

      npm install express

      ```

9. **\*\*Compare Traditional web server model and Node.js Process model.\*\***

    - Traditional web server model uses multiple threads to handle requests, while Node.js uses a single-threaded event loop.

10. **\*\*Write a Program which uses addListener() method of EventEmitter class.\*\***

    ```js

    const EventEmitter = require('events');

    const emitter = new EventEmitter();

    emitter.addListener('event', () => {

        console.log('An event occurred!');

    });

    emitter.emit('event');

    ```

11. **\*\*Write a Program to update table records using Node.js and MySQL database.\*\***

    ```js

    const mysql = require('mysql');

    const connection = mysql.createConnection({

        host: 'localhost',

        user: 'root',

        password: '',

        database: 'test'

    });

    connection.connect();

    connection.query('UPDATE users SET name = "John" WHERE id = 1', (err, result) => {

        if (err) throw err;

        console.log('Record updated');

    });

    connection.end();

    ```

12. **\*\*Explain Node.js Process Model with the help of a diagram.\*\***

    - Node.js uses a single-threaded event loop model to handle multiple clients. The event loop processes incoming requests and delegates blocking operations to worker threads.

13. **\*\*How does Node.js handle a file request?\*\***

    - Node.js uses the `fs` module to handle file requests. It reads the file asynchronously and sends the content back to the client.

14. **\*\*What is the Purpose of object module exports in Node.js?\*\***

    - `module.exports` is used to export functions, objects, or primitives from a module so they can be used in other modules.

15. **\*\*What are the advantages of Node.js?\*\***

    - Non-blocking I/O

    - Single-threaded event loop

    - Fast execution

    - Large ecosystem of packages

16. **\*\*How to write asynchronous data to a file? Explain with a suitable example.\*\***

    ```js

    const fs = require('fs');

    fs.writeFile('example.txt', 'Hello, world!', (err) => {

        if (err) throw err;

        console.log('File has been saved!');

    });

    ```

17. **\*\*Write a program which uses addListener() method of EventEmitter class.\*\***

    ```js

    const EventEmitter = require('events');

    const emitter = new EventEmitter();

    emitter.addListener('event', () => {

        console.log('An event occurred!');

    });

    emitter.emit('event');

    ```

18. **\*\*Write a short note on NPM.\*\***

    - NPM (Node Package Manager) is a package manager for JavaScript. It allows developers to share and reuse code, and it manages dependencies for Node.js projects.

19. **\*\*Write a code for selecting all records from Player's table.\*\***

    ```js

    const mysql = require('mysql');

    const connection = mysql.createConnection({

        host: 'localhost',

        user: 'root',

        password: '',

        database: 'test'

    });

    connection.connect();

    connection.query('SELECT \* FROM players', (err, results) => {

        if (err) throw err;

        console.log(results);

    });

    connection.end();

    ```

20. **\*\*Explain module.exports in Node.js.\*\***

    - `module.exports` is used to export functions, objects, or primitives from a module so they can be used in other modules.

21. **\*\*Compare Traditional web server Model and Node.js process Model.\*\***

    - Traditional web server model uses multiple threads to handle requests, while Node.js uses a single-threaded event loop.

22. **\*\*Write Node.js application to create user-defined square module to find the area of square & display the details on console.\*\***

    ```js

    // square.js

    module.exports.area = (side) => side \* side;

    // app.js

    const square = require('./square');

    console.log(`Area of square: ${square.area(5)}`);

    ```

23. **\*\*What is web server?\*\***

    - A web server is a server that handles HTTP requests and serves web pages to clients. In Node.js, you can create a web server using the `http` module.

24. **\*\*Write a program to create a file in Node.js.\*\***

    ```js

    const fs = require('fs');

    fs.writeFile('example.txt', 'Hello, world!', (err) => {

        if (err) throw err;

        console.log('File has been saved!');

    });

    ```

25. **\*\*Explain parameters of create connection.\*\***

    - `host`: The hostname of the database.

    - `user`: The username to connect to the database.

    - `password`: The password to connect to the database.

    - `database`: The name of the database to connect to.

26. **\*\*Explain Event-driven programming.\*\***

    - Event-driven programming is a programming paradigm where the flow of the program is determined by events such as user actions, sensor outputs, or messages from other programs.

27. **\*\*What is Anonymous function?\*\***

    - An anonymous function is a function without a name. It is often used as an argument to other functions or as an immediately invoked function expression (IIFE).

28. **\*\*Explain is module?\*\***

    - A module is a reusable piece of code that can be imported and used in other parts of an application.

29. **\*\*Explain the use of Buffer and How to create Buffer.\*\***

    - Buffers are used to handle binary data in Node.js. You can create a buffer using:

      ```js

      const buffer = Buffer.alloc(size);

      ```

30. **\*\*Write down the steps to create a local module.\*\***

    - Create a new file for the module.

    - Define the functions or objects you want to export.

    - Use `module.exports` to export them.

    - Import the module in another file using `require()`.

31. **\*\*Write Node.js program which will convert the output “SY BCA” into upper-case.\*\***

    ```js

    const str = "SY BCA";

    console.log(str.toUpperCase());

    ```

32. **\*\*Write a Node.js Script to check a given number is Perfect or Not using function.\*\***

    ```js

    function isPerfect(num) {

        let sum = 0;

        for (let i = 1; i < num; i++) {

            if (num % i === 0) sum += i;

        }

        return sum === num;

    }

    console.log(isPerfect(6)); // Output: true

    ```

33. **\*\*What is Module? Explain Third Party Module.\*\***

    - A module is a reusable piece of code that can be imported and used in other parts of an application. Third-party modules are modules created by the community and published on the NPM registry.

34. **\*\*What is Synchronous and Asynchronous approach?\*\***

    - Synchronous approach executes tasks sequentially, blocking the execution of subsequent tasks until the current task is completed. Asynchronous approach allows tasks to be executed concurrently, without blocking the execution of subsequent tasks.

35. **\*\*What is Event Driven Programming?\*\***

    - Event-driven programming is a programming paradigm where the flow of the program is determined by events such as user actions, sensor outputs, or messages from other programs.

36. **\*\*Explain Anonymous function with an example.\*\***

    - An anonymous function is a function without a name. Example:

      ```js

      const sum = function(a, b) {

          return a + b;

      };

      console.log(sum(2, 3)); // Output: 5

      ```

37. **\*\*Explain the syntax to create a text file and delete the file.\*\***

    - Create a file:

      ```js

      const fs = require('fs');

      fs.writeFile('example.txt', 'Hello, world!', (err) => {

          if (err) throw err;

          console.log('File has been saved!');

      });

      ```

    - Delete a file:

      ```js

      const fs = require('fs');

      fs.unlink('example.txt', (err) => {

          if (err) throw err;

          console.log('File has been deleted!');

      });

      ```

38. **\*\*Explain any two methods of EventEmitter class:\*\***

    - `on(event, listener)`: Adds a listener for the specified event.

    - `emit(event, [arg1], [arg2], [...])`: Emits the specified event, calling all listeners with the provided arguments.

39. **\*\*Write a Node.js Script to check a given number is even or odd using function.\*\***

    ```js

    function isEven(num) {

        return num % 2 === 0;

    }

    console.log(isEven(4)); // Output: true

    ```

40. **\*\*Write Node.js program to count the occurrence of a given word in a file and display the count on Console.\*\***

    ```js

    const fs = require('fs');

    const filePath = 'example.txt';

    const word = 'hello';

    fs.readFile(filePath, 'utf8', (err, data) => {

        if (err) throw err;

        const count = data.split(word).length - 1;

        console.log(`The word "${word}" occurs ${count} times.`);

    });

    ```

41. **\*\*Write a Program to define Module Circle.js which exports the functions area() and circumference() and display the details on console.\*\***

    ```js

    // circle.js

    module.exports.area = (radius) => Math.PI \* radius \* radius;

    module.exports.circumference = (radius) => 2 \* Math.PI \* radius;

    // app.js

    const circle = require('./circle');

    console.log(`Area: ${circle.area(5)}`);

    console.log(`Circumference: ${circle.circumference(5)}`);

    ```

42. **\*\*Which databases does Node.js support?\*\***

    - MySQL

    - MongoDB

    - PostgreSQL

    - SQLite

43. **\*\*Write down the Connection String of Node.js and MySQL.\*\***

    ```js

    const mysql = require('mysql');

    const connection = mysql.createConnection({

        host: 'localhost',

        user: 'root',

        password: '',

        database: 'test'

    });

    ```

44. **\*\*Explain the use of REPL.\*\***

    - REPL (Read-Eval-Print Loop) is an interactive shell that processes Node.js expressions. It reads the input, evaluates it, prints the result, and loops back to read the next input.