Multiple Choice Answers:

1. B) Ryan Dahl
2. d. All of the above
3. b. Asynchronous
4. c. Runtime environment
5. D) JavaScript

Identification:

1. console.log("Listening on port 3000");
2. app.listen(3000);
3. Hello World
4. Parameters
5. Reassigned and redeclared
6. const path = require('path');
7. http.createServer()
8. console.log("Hello World!");
9. const fs = require('fs');
10. Request and Response objects in HTTP server functions

True or False:

1. False
2. True
3. False
4. True
5. True
6. True
7. True
8. False
9. True
10. False

Enumeration / Essay

1. **Uses of NPM, Give at least ten (10) of them**
2. **Installing Packages**
3. **Installing All Dependencies**
4. **Updating Packages**
5. **Versioning**
6. **Running Tasks**
7. **Uninstalling packages**
8. **Share code**
9. Installing Global Packages
10. Checking Outdated Packages
11. Uninstalling Packages
12. **How Node architecture works**

Node uses an event-driven, non-blocking I/O model. It relies on single-threaded processing to handle multiple connections concurrently, allowing it to manage high input-output demands without requiring multiple threads. This makes Node ideal for data-intensive, real-time applications.

1. **What's special about Node**

Node's ability to handle asynchronous operations, its efficient handling of data-intensive tasks, and compatibility with JavaScript make it an attractive choice for server-side scripting, particularly for real-time applications like chat apps, online games, and collaborative tools.

1. **What is Node**

Node.js is a JavaScript runtime built on Chrome's V8 engine, allowing developers to run JavaScript on the server side. It's designed for developing scalable network applications.

1. **Why Node is needed in development**

Node’s non-blocking, asynchronous I/O processing is ideal for building real-time applications. It allows developers familiar with JavaScript to use the same language on both the client and server, streamlining development for faster, more maintainable projects.