Portfolio (Week-3)

This week’s lab session focused on **JavaScript functions** and working with JavaScript modules. We used **Visual Studio Code** for all coding exercises, ensuring a structured approach to learning. Here is how the week unfolded:

1. **Checking Node.js Installation** Before beginning the exercises, I verified whether Node.js was installed on my system. I ran the command:

This command displayed the installed version of Node.js, confirming that the environment was ready for coding.

A screenshot of a computer

Description automatically generated

**Creating the Exercise File** To start, I created a new JavaScript file in my Week 3 folder. This file was used to write and execute the lab exercises for this week. Organizing files in dedicated folders helps maintain clarity and structure in my work.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Exercise 2: Creating an Arrow Function**

In this exercise, I created an arrow function to perform a specific task. Arrow functions offer a concise syntax for writing functions in JavaScript. They are particularly useful for simplifying callback functions and maintaining lexical this binding.

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Exercise 3: Working with JavaScript Modules**

This exercise involved creating and connecting multiple JavaScript files to demonstrate modular programming. Modular programming helps in organizing code into reusable and maintainable components. Here are the steps I followed:

1. **Creating Files**
   * I created three separate files: index.js, person.js, and student.js.
   * Each file had a distinct purpose and functionality.
2. Writing the StudentInfo.js File

The StudentInfo.js file contained information specific to students, such as their name, age, and grade.

A screenshot of a computer program

Description automatically generated

**Writing the Person.js File** The Person.js file acted as a base module, defining general properties and methods applicable to all persons.

A screenshot of a computer

Description automatically generated

**Connecting Modules in Index.js** Finally, I wrote additional lines of code in index.js to import and utilize the functionality defined in StudentInfo.js and Person.js. This demonstrated how JavaScript modules could be interconnected to build complex applications.

A screenshot of a computer program

Description automatically generated

**Final Output** After running the index.js file, the program produced the expected output, validating that the modules were successfully connected and functional.

A screenshot of a computer program

Description automatically generated

Exercise 4:

A screen shot of a computer

Description automatically generated

**Learning Objectives**

* To understand the basics of JavaScript functions, including arrow functions.
* To explore modular programming using JavaScript modules.
* To learn how to connect and utilize multiple JavaScript files effectively.

**Key Points**

* Modular programming enhances code reusability and maintainability.
* Arrow functions simplify syntax and help maintain the lexical scope of this.
* Organizing files and folders is crucial for structured and efficient coding.

**Problems Faced**

1. **Syntax Errors in Arrow Functions:** While writing the arrow function, I encountered syntax errors due to incorrect usage of parentheses and curly braces.
   * **Solution:** I reviewed the syntax rules for arrow functions and referred to online resources to resolve the issue.
2. **Module Import Issues:** When connecting the modules, I faced errors where the imported functions were not recognized.
   * **Solution:** I double-checked the file paths and ensured that the exported functions and variables were properly defined in the source files.
3. **Error Running index.js:** Initially, running the index.js file produced unexpected results due to incorrect sequencing of imports.
   * **Solution:** I reorganized the imports to ensure that dependencies were loaded in the correct order.

By overcoming these challenges, I gained confidence in working with JavaScript functions and modules. This lab session helped enhance my problem-solving skills and deepened my understanding of JavaScript fundamentals.