|  |  |  |
| --- | --- | --- |
| **Name of Student:** Atharva Mohite | | |
| **Roll Number:** 34 | | **LAB Assignment Number:** 8 |
| **Title of LAB Assignment:** TypeScript installation, Environment Setup, Programs on decision making/functions/class & object) | | |
| **DOP:** 11-10-2023 | | **DOS:** 20-10-2023 |
| **CO Mapped:** CO4 | **PO Mapped:** PO3, PO5, PSO1, PSO2 | **Signature:** |

# 

# PRACTICAL 7

## Aim: TypeScript installation, Environment Setup, Programs on decision making/functions/class & object)

## Theory:

Setting up a TypeScript development environment and writing programs that cover decision making, functions, and classes and objects can be done by following these steps. I'll provide a step-by-step guide for each part of your request.

**1. TypeScript Installation and Environment Setup:**

To set up your TypeScript environment, you need Node.js and npm (Node Package Manager) installed on your machine. If you haven't already, follow these steps to set up TypeScript:

**a. Install Node.js and npm:**

Go to the official Node.js website (<https://nodejs.org/>) and download the LTS version for your platform. Once downloaded, run the installer and follow the instructions.

**b. Install TypeScript:**

Open your terminal or command prompt and run the following command to install TypeScript globally:

npm install -g typescript

**2. Writing Programs:**

Now that you have TypeScript installed, you can start writing programs. Create a file with an extension of “.ts”. Eg. practical 8.ts

Write a program of your choice.

**3. Compiling and Running TypeScript:**

After writing your TypeScript code, you need to compile it to JavaScript using the TypeScript Compiler (tsc). In your terminal, navigate to the directory where your TypeScript files are located and run:

tsc practical 8.ts

This will generate a JavaScript file with the same name (e.g., your-file.js).

You can then execute the JavaScript program using Node.js:

node practical 8.js

## Aim: Programs on decision making/functions/class & object)

// decision making

let age: number = 20;

if (age >= 18) {

console.log("You are an adult.");

}

else {

console.log("You are a minor.");

}

// functions

function add(a: number, b: number): number {

return a + b;

}

let result: number = add(5, 3);

console.log("Sum:", result);

// classes

class Car {

make: string;

model: string;

constructor(make: string, model: string) {

this.make = make;

this.model = model;

}

display(): void {

console.log('Car: ', this.make, this.model);

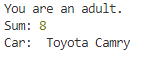
}

}

let myCar = new Car("Toyota", "Camry");

myCar.display();

**Output**

****

## Conclusion: We learnt about working with typescript in Nodejs.