Installing and Setting Up your TypeScript Environment

Hello, welcome to the first lab session for SSUI! Many (if not all) of the programming assignments we'll have throughout the semester will be in TypeScript, which is a variant to JavaScript.

This handout is a guide for walking through setting up a VSCode environment to support development in TypeScript.

For this course, using VSCode will the be the only officially supported method for development. While Scott and I will make attempts to assist you if you choose to use alternate IDEs or methods, due to time constraints and general unfamiliarity we may not be able to help you with your specific setup.

1. **Install Visual Studio Code:** If you haven't installed VSCode yet, you can download it from https://code.visualstudio.com/download. VSCode is a popular code editor and it's great for TypeScript development.
2. **Install Node.js and npm:** TypeScript is a Node.js application, so you need to have Node.js and npm (Node Package Manager) installed on your system. You can download Node.js and npm from https://nodejs.org/en/download/. After installation, you can check the installed versions by running `node -v` and `npm -v` in your terminal/command prompt.
3. **Install TypeScript:** After installing Node.js and npm, you can install TypeScript globally on your system by running `npm install -g typescript` in your terminal/command prompt. You can check the installed TypeScript version by running `tsc -v`.
4. **Create a New TypeScript Project:**
   - Create a new directory for your project and open it in VSCode.
   - Open the terminal in VSCode by clicking on `View -> Terminal`.
   - *If not provided:* Run `npm init -y` to create a new `package.json` file with default values.
   - Run `tsc --init` to create a new `tsconfig.json` file. This file is used to configure how the TypeScript compiler will convert your TypeScript code into JavaScript. (*Note: We will provide a `tsconfig.json` which should work for your purposes for P1. However, if you are setting up from scratch for a new project, consider Step 5.)
5. **Configure tsconfig.json (if setting up from scratch):** open the `tsconfig.json` file and make sure it has the following settings:

```
{
  "compilerOptions": {
    "target": "es2015",
    "module": "commonjs",
    "outDir": "./out",
    "rootDir": ".",
    "strict": true
  }
}
```

   - `target`: Specifies the ECMAScript (or JavaScript) target version.
   - `module`: Specifies the module system.
   - `outDir`: Specifies the output directory for the compiled JavaScript files.
   - `rootDir`: Specifies the root directory of your TypeScript files.
   - `strict`: Enables all strict type-checking options.

6. **Create TypeScript Code:**
    - Create a new TypeScript file (e.g., `p1.ts`) and write some TypeScript code. Make sure the new file is in the same directory specified in `rootDir`
7. **Compile TypeScript Code:**
    - Run `tsc` in the terminal. This will compile your TypeScript code into JavaScript and output the JavaScript files in the `out` directory.
8. **Run JavaScript Code:**
    - Run `node out/p1.js` in the terminal. This will run your JavaScript code in Node.js.
    - Alternatively, for projects which involve projects run in a browser, launch the project by running the corresponding HTML file, (for `p1`, open `index.html` in a browser, preferably Chrome or Firefox).
9. **Debug TypeScript in VSCode (Optional):**
    - Click on the Run view (`View -> Run`).
    - Click on `create a launch.json file`.
    - Select `Node.js`.
    - Update the `program` field in the `launch.json` file to point to the JavaScript file in the `out` directory.
    - Click on the Start Debugging button or press F5 to start debugging.

That's it! You have now set up VSCode for running and building TypeScript projects.