Name: Allan Rodrigues
Class: TE IT A
Roll no: 59
Pid:191104

St. Francis Institute of Technology, Mumbai-400 103

Department Of Information Technology

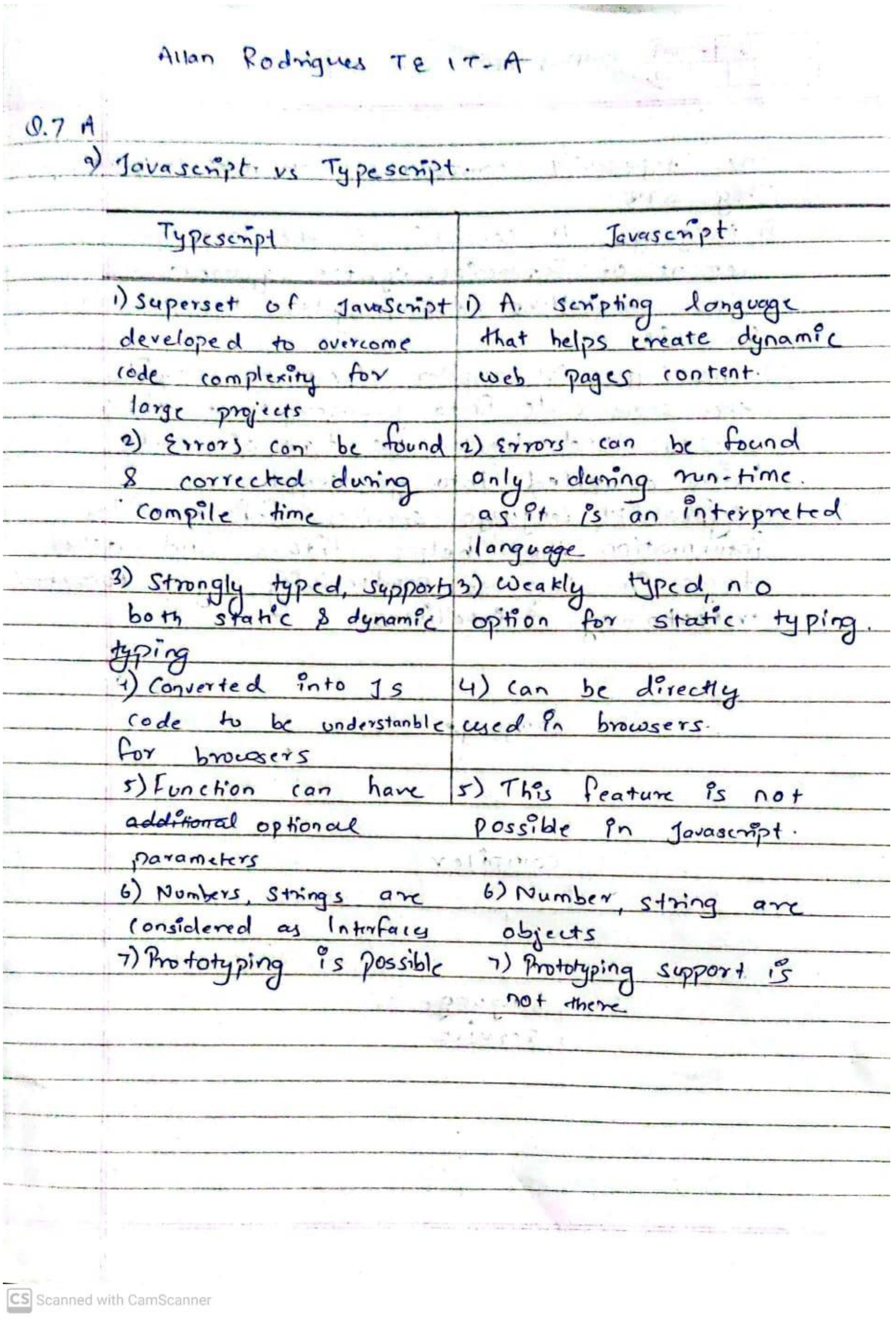
A.Y. 2021-2022
Class: TE-ITA/B, Semester: VI

Subject: **Web Lab**

Experiment – 2: To study the basics of TypeScript through small code snippets.

1. **Aim:** To study simple program in TypeScript compile and run it.
2. **Objectives:** Aim of this experiment is that, the students will be able
   - To know Installing TypeScript
   - Read and understand commonly-used TypeScript syntax and patterns
   - to understand TypeScript error handling, Looping
3. **Outcomes:** After study of this experiment, the students will be able
   - To install Typescript.
   - Write code, compile and execute the code to get result.
   - To know syntaxes for loop constructs, defining variables, printing on screen, taking input from user
4. **Prerequisite:** Basic knowledge of JavaScript is required
5. **Requirements:** Personal Computer, Windows operating system, VSCode editor, browser, Internet Connection, node, typescript compiler, google doc.
6. **Pre-Experiment Exercise:**
   Brief Theory: Refer shared material
7. **Laboratory Exercise**
   A. **Procedure:**
      a. **Answer the following:**

         - Javascript vs Typescript.



Allan Rodrigues TE IT-A

Q.7 A

a) Javascript vs Typescript.

Typescript	Javascript
1) Superset of JavaScript developed to overcome code complexity for large projects	1) A scripting language that helps create dynamic web pages content.
2) Errors can be found & corrected during compile time	2) Errors can be found only during run-time as it is an interpreted language
3) Strongly typed, supports both static & dynamic typing	3) Weakly typed, no option for static typing.
4) Converted into JS code to be understanble for browsers	4) Can be directly used in browsers.
5) Function can have ~~additional~~ optional parameters	5) This feature is not possible in Javascript.
6) Numbers, Strings are considered as Interfaces	6) Number, string are objects
7) Prototyping is possible	7) Prototyping support is not there.

- How to use Typescript?

1. Install the TypeScript compiler

To start off, the TypeScript compiler will need to be installed in order to convert TypeScript files into JavaScript files. To do this, TypeScript can either be installed globally (available anywhere in your file system) or locally (only available at the project level).

2. Make sure your editor is setup to support TypeScript

You'll want to make sure your editor is properly configured to work with TypeScript. For example, you may need to install a plugin (such as atom-typescript if using the atom editor), in order to fully take advantage of TypeScript in your editor. If using VS Code, TypeScript support is built-in, so there are no extensions required.

3. Create a tsconfig.json file

A tsconfig.json file is used to configure TypeScript project settings. The tsconfig.json file should be put in the project's root directory. The file allows you to configure the TypeScript compiler with different options. You can have the tsconfig.json contain an empty JSON object if you just want to get TypeScript to work, but if you need the TypeScript compiler to behave differently (such as output transpiled JavaScript files in a specific output directory), you can read more about which settings can be configured.

4. Transpile TypeScript to JavaScript

In order to transpile your TypeScript code to JavaScript, the tsc command needs to be run in the terminal. Running tsc will have the TypeScript compiler search for the tsconfig.json file which will determine the project's root directory as well as which options to use when compiling the TypeScript and transpiling .ts files to .js files.

To quickly test that the setup works, you can create a test TypeScript file and then run tsc in the command line and see if a JavaScript file is generated beside the TypeScript file.

- TypeScript Features?

Object-Oriented language: TypeScript provides a complete feature of an object-oriented programming language such as classes, interfaces, inheritance, modules, etc. In TypeScript, we can write code for both client-side as well as server-side development.

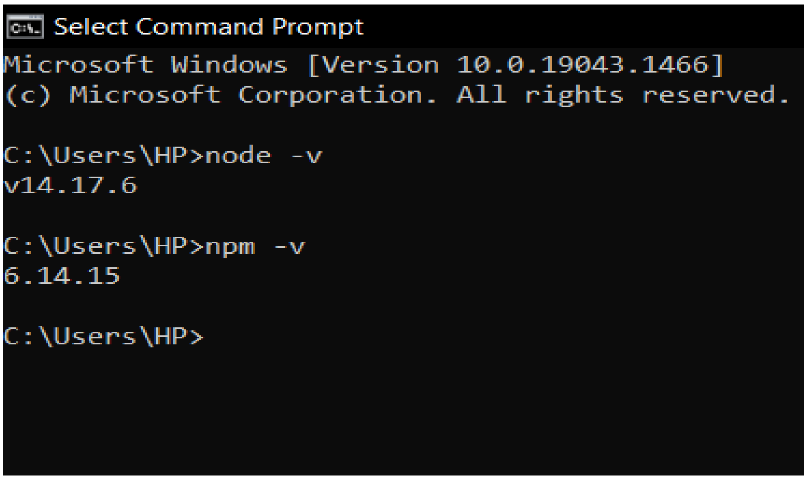
TypeScript supports JavaScript libraries: TypeScript supports each JavaScript elements. It allows the developers to use existing JavaScript code with the TypeScript. Here, we can use all of the JavaScript frameworks, tools, and other libraries easily.

JavaScript is TypeScript: It means the code written in JavaScript with valid .js extension can be converted to TypeScript by changing the extension from .js to .ts and compiled with other TypeScript files.

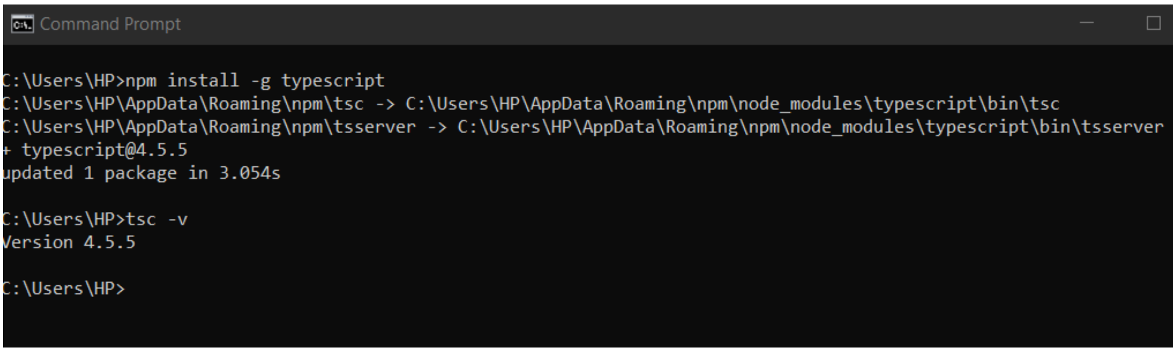
TypeScript is portable: TypeScript is portable because it can be executed on any browsers, devices, or any operating systems. It can be run in any environment where JavaScript runs on. It is not specific to any virtual-machine for execution.

b. Attach screenshots:

- Typescript installation

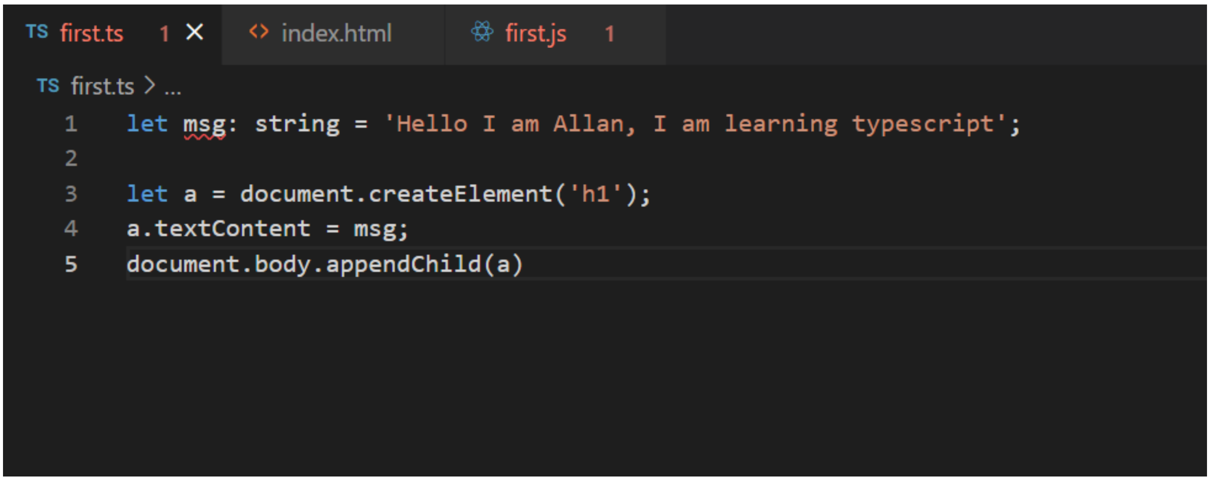


```
Microsoft Windows [Version 10.0.19043.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>node -v
v14.17.6

C:\Users\HP>npm -v
6.14.15

C:\Users\HP>
```

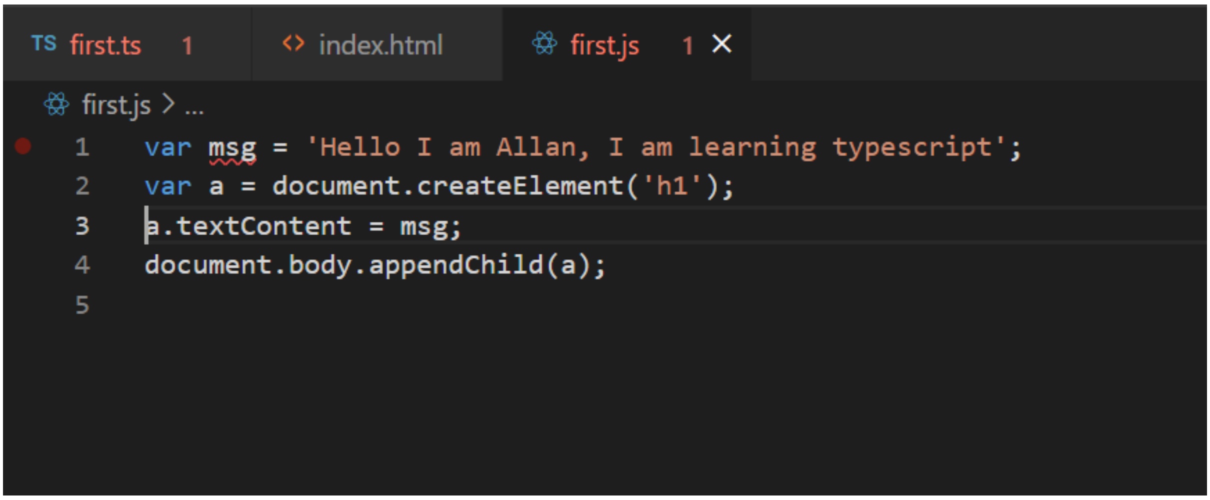


```
C:\Users\HP>npm install -g typescript
C:\Users\HP\AppData\Roaming\npm\tsc -> C:\Users\HP\AppData\Roaming\npm\node_modules\typescript\bin\tsc
C:\Users\HP\AppData\Roaming\npm\tsserver -> C:\Users\HP\AppData\Roaming\npm\node_modules\typescript\bin\tsserver
+ typescript@4.5.5
updated 1 package in 3.054s

C:\Users\HP>tsc -v
Version 4.5.5

C:\Users\HP>
```

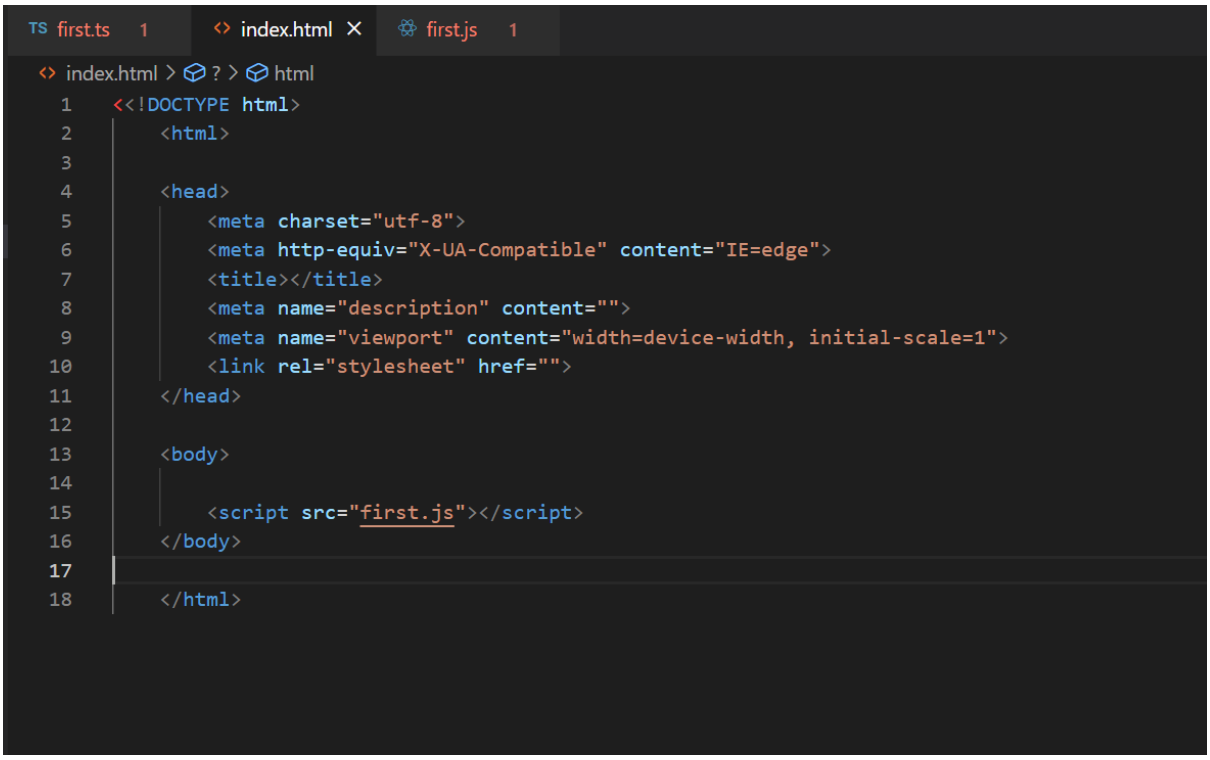
- Typescript Program code and output



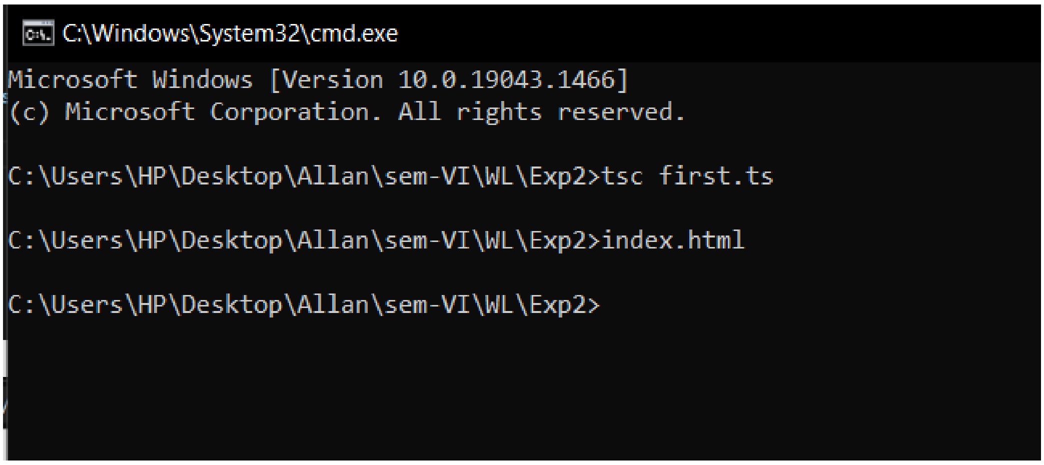
```
let msg: string = 'Hello I am Allan, I am learning typescript';

let a = document.createElement('h1');
a.textContent = msg;
document.body.appendChild(a)
```



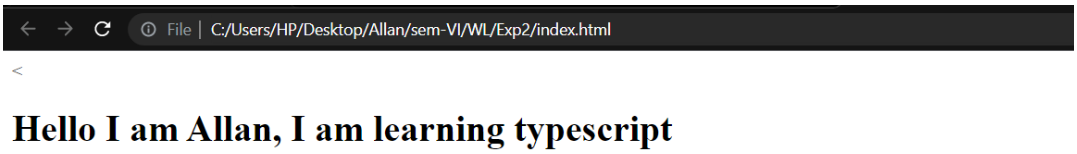
```
var msg = 'Hello I am Allan, I am learning typescript';
var a = document.createElement('h1');
a.textContent = msg;
document.body.appendChild(a);

```



```
<<!DOCTYPE html>
    <html>

    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <title></title>
        <meta name="description" content="">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel="stylesheet" href="">
    </head>

    <body>

        <script src="first.js"></script>
    </body>

    </html>
```

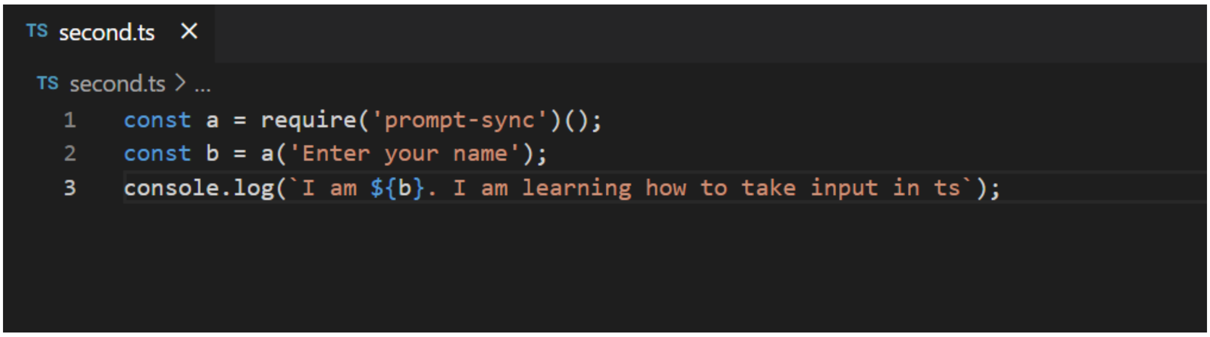


```
Microsoft Windows [Version 10.0.19043.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP\Desktop\Allan\sem-VI\WL\Exp2>tsc first.ts

C:\Users\HP\Desktop\Allan\sem-VI\WL\Exp2>index.html

C:\Users\HP\Desktop\Allan\sem-VI\WL\Exp2>
```

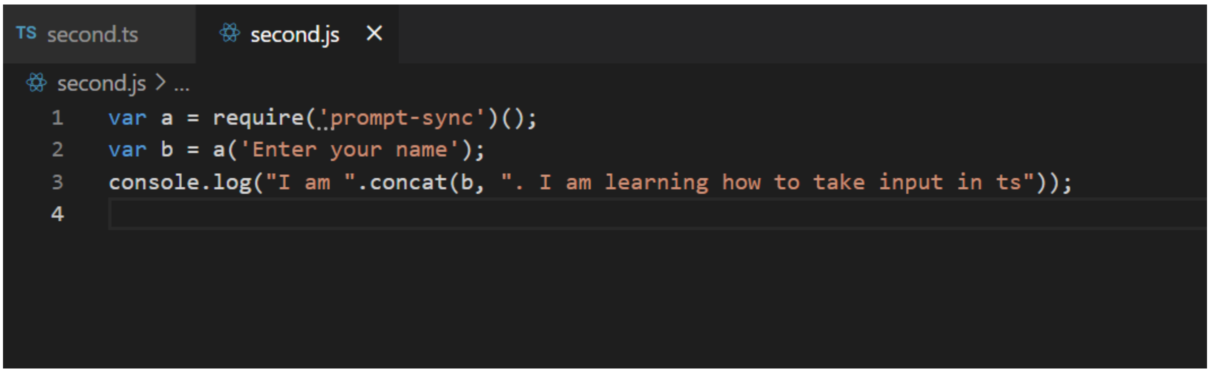


<

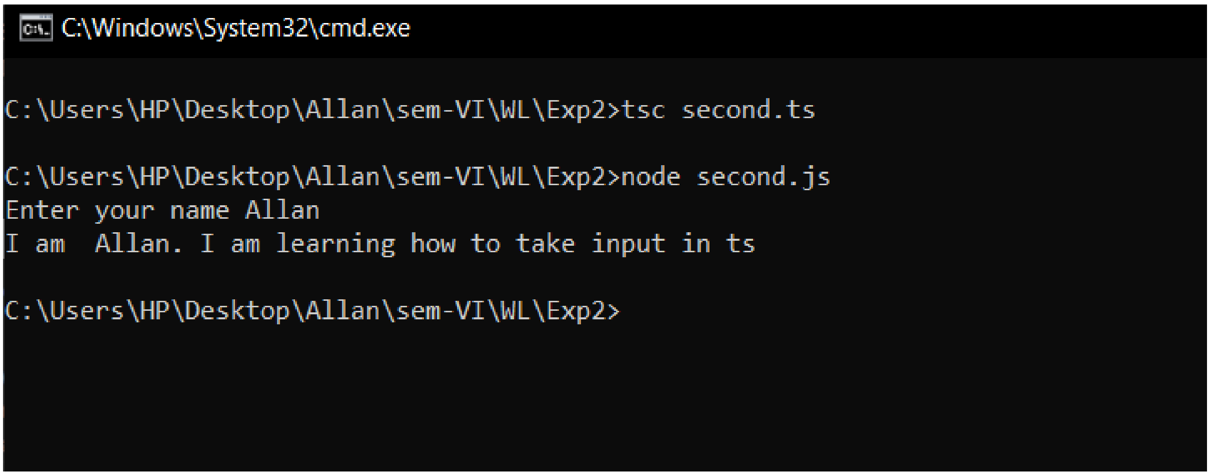
Hello I am Allan, I am learning typescript



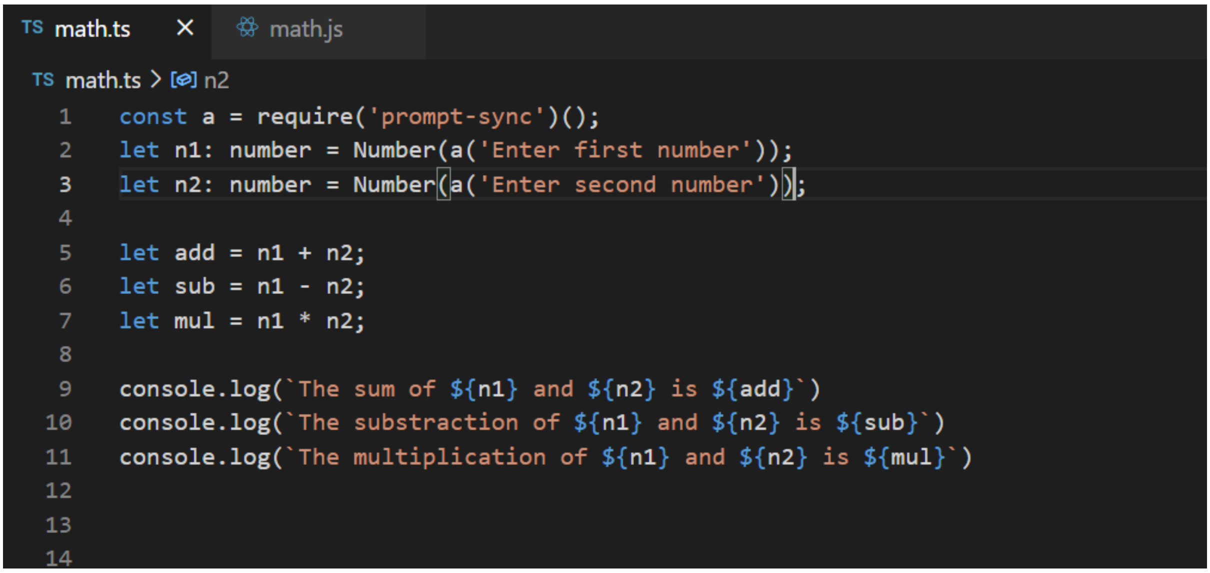
```
const a = require('prompt-sync')();
const b = a('Enter your name');
console.log(`I am ${b}. I am learning how to take input in ts`);
```



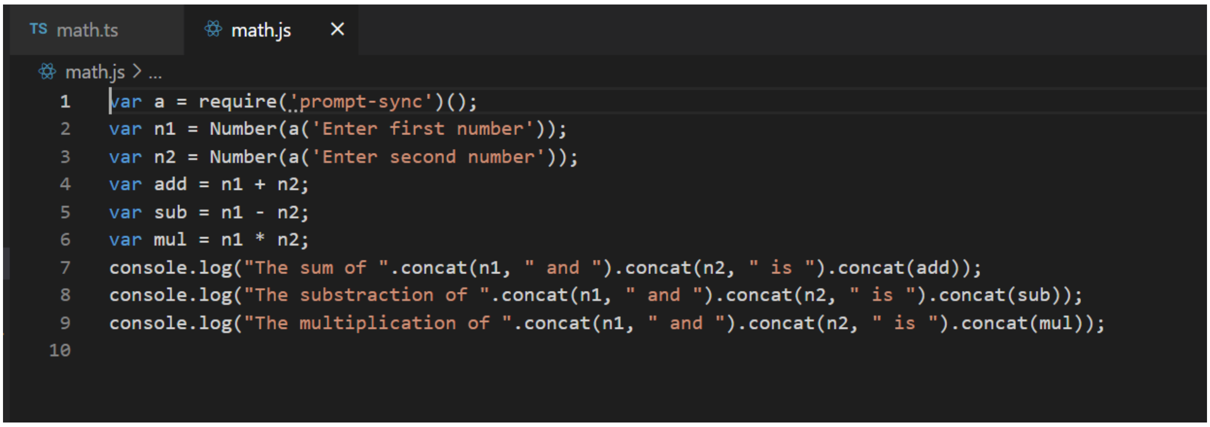
```
var a = require('prompt-sync')();
var b = a('Enter your name');
console.log("I am ".concat(b, ". I am learning how to take input in ts"));

```



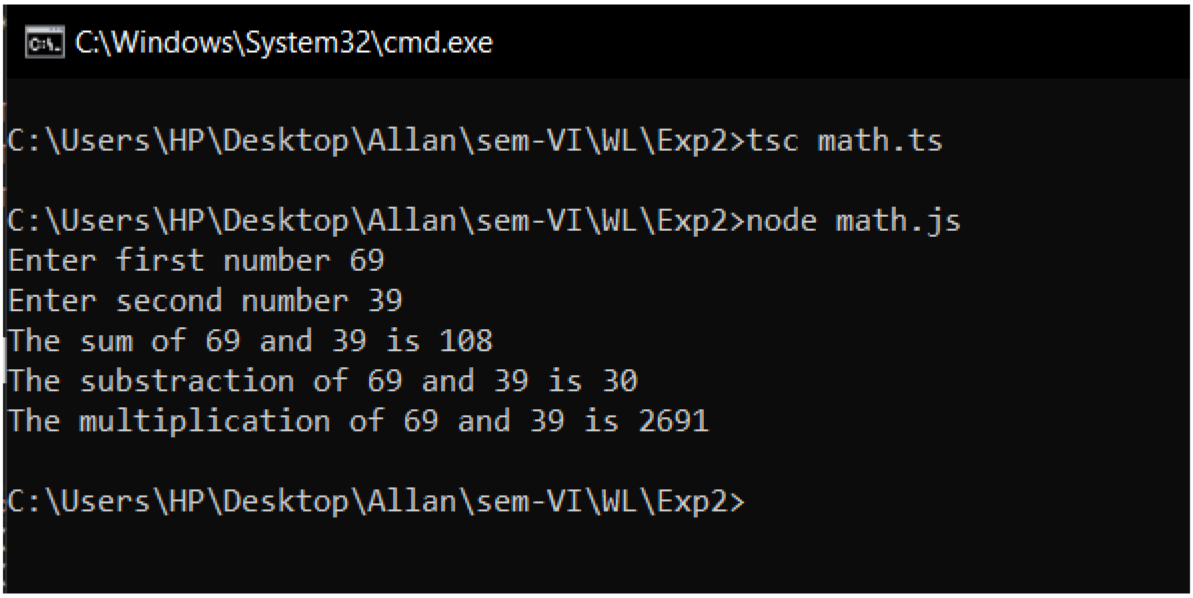
```
C:\Users\HP\Desktop\Allan\sem-VI\WL\Exp2>tsc second.ts

C:\Users\HP\Desktop\Allan\sem-VI\WL\Exp2>node second.js
Enter your name Allan
I am  Allan. I am learning how to take input in ts

C:\Users\HP\Desktop\Allan\sem-VI\WL\Exp2>
```



```
const a = require('prompt-sync')();
let n1: number = Number(a('Enter first number'));
let n2: number = Number(a('Enter second number'));

let add = n1 + n2;
let sub = n1 - n2;
let mul = n1 * n2;

console.log(`The sum of ${n1} and ${n2} is ${add}`)
console.log(`The substraction of ${n1} and ${n2} is ${sub}`)
console.log(`The multiplication of ${n1} and ${n2} is ${mul}`)
```



```
var a = require('prompt-sync')();
var n1 = Number(a('Enter first number'));
var n2 = Number(a('Enter second number'));
var add = n1 + n2;
var sub = n1 - n2;
var mul = n1 * n2;
console.log("The sum of ".concat(n1, " and ").concat(n2, " is ").concat(add));
console.log("The substraction of ".concat(n1, " and ").concat(n2, " is ").concat(sub));
console.log("The multiplication of ".concat(n1, " and ").concat(n2, " is ").concat(mul));
```



```
C:\Users\HP\Desktop\Allan\sem-VI\WL\Exp2>tsc math.ts

C:\Users\HP\Desktop\Allan\sem-VI\WL\Exp2>node math.js
Enter first number 69
Enter second number 39
The sum of 69 and 39 is 108
The substraction of 69 and 39 is 30
The multiplication of 69 and 39 is 2691

C:\Users\HP\Desktop\Allan\sem-VI\WL\Exp2>
```

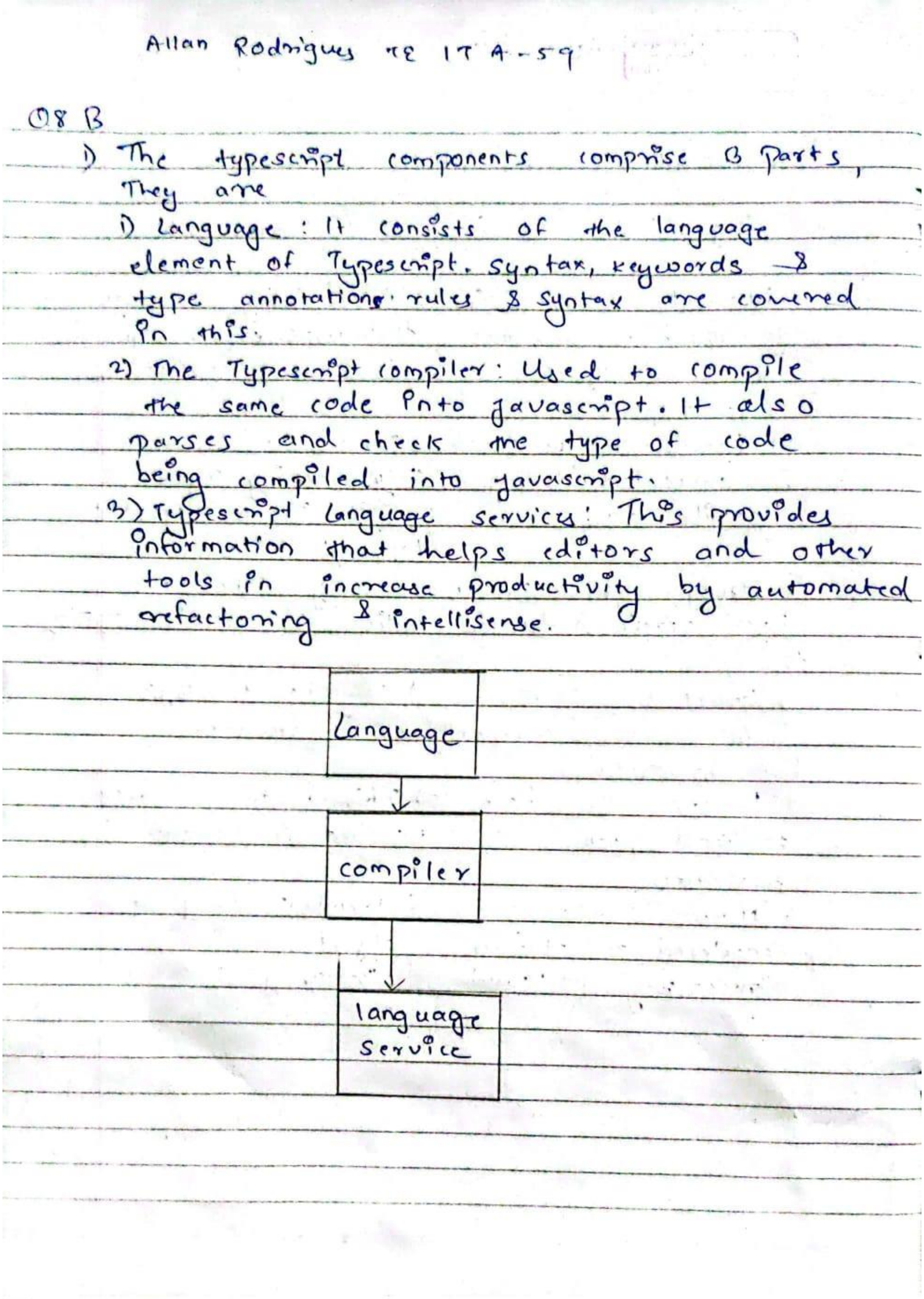
8. **Post-Experiments Exercise**

A. **Extended Theory:**

Nil

B. **Questions:**

- Draw and explain Components of TypeScript.



Allan Rodrigues TE IT A-59

Q8 B

1) The typescript components comprise 3 parts, They are

1) Language : It consists of the language element of Typescript. syntax, keywords & type annotations rules & syntax are covered in this.

2) The Typescript compiler : Used to compile the same code into javascript. It also parses and check the type of code being compiled into javascript.

3) Typescript language services : This provides information that helps editors and other tools in increase productivity by automated refactoring & intellisense.

- List out the built-in data types in TypeScript.

- Boolean
  The boolean type represents true and false values. It is one of the primitive data types in TypeScript. Its corresponding JavaScript primitive type is boolean and its JavaScript constructor is Boolean.

- Number
  The number type represents all real numbers (integers and floating-point numbers). It is one of the primitive data types in TypeScript. Its corresponding JavaScript primitive type is number and its JavaScript constructor is Number.

- string
  The string type represents textual data. It is one of the primitive data types in TypeScript. Its corresponding JavaScript primitive type is string and its JavaScript constructor is String.

- undefined
  The undefined type represents only the undefined value. It is one of the primitive data types in TypeScript. Its corresponding JavaScript primitive type is undefined. A value of type undefined can only be presented using literal notation since it doesn't have a constructor.

- null
  The null type represents only the null value. It is one of the primitive data types in TypeScript. Though it's a JavaScript primitive value, its runtime type is object which is a bug in JavaScript. A value of type null can only be presented using literal notation since it doesn't have a constructor.

- void
  The void type represents a value without any type. It is one of the primitive data types in TypeScript. This type purely exists in the TypeScript realm since it does not represent an actual value.

- never
  The never type represents a value that never exists. It is one of the primitive data types in TypeScript. This type purely exists in the TypeScript realm since it does not represent an actual value.

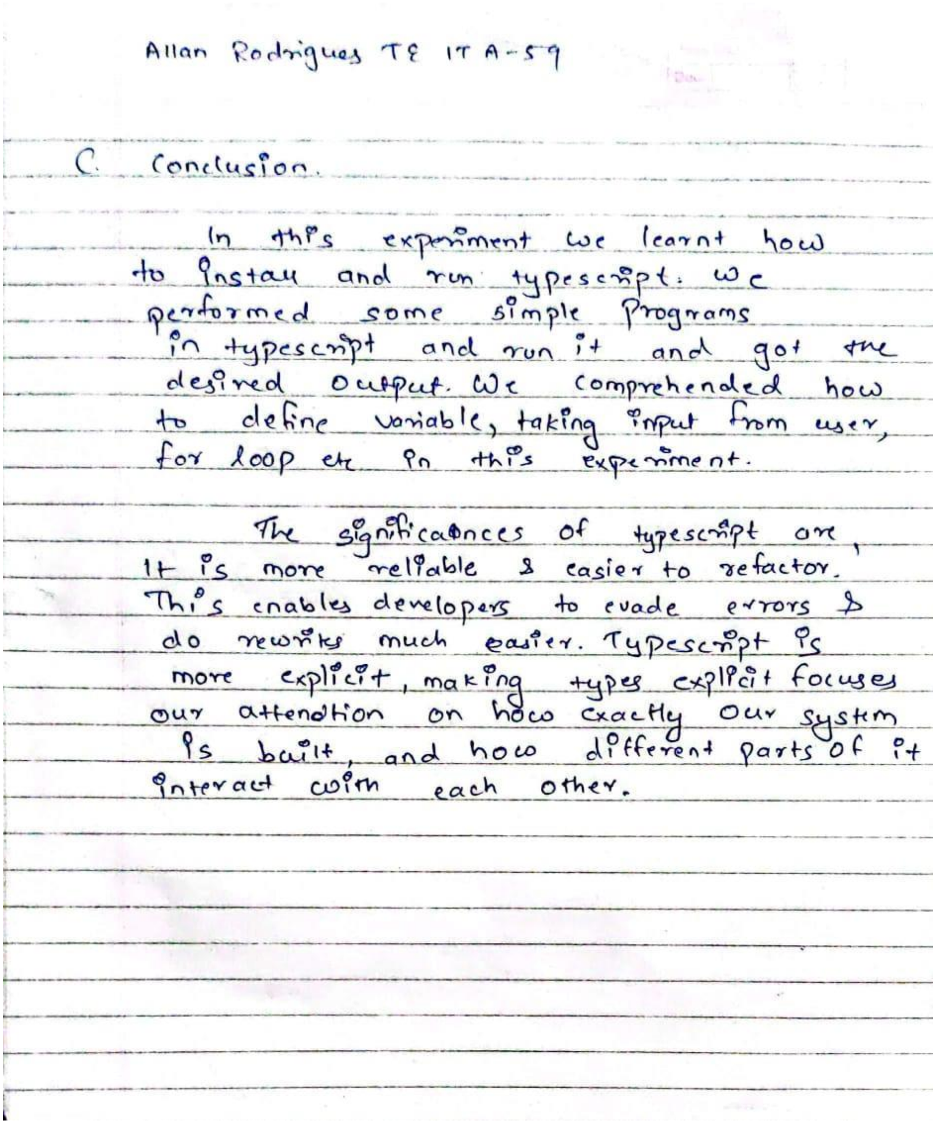
- any
  The any type is a special type in TypeScript since it can represent virtually all possible values. It is one of the primitive data types in TypeScript. This type purely exists in the TypeScript realm since it doesn't have a fixed data type at the runtime.

- unknown
  The unknown type is another top type in TypeScript which acts exactly similar to any with a few minor differences. It is one of the primitive data types in TypeScript. This type purely exists in the TypeScript realm since it doesn't have a fixed data type at the runtime.

- Array
  The Array type represents a list of items of a given data type. Since Array is composed of items of primitive or abstract types, it is an abstract type.

C. Conclusion:(write in hand, scan and paste)

- Write what was performed in the experiment.
- Write the significance of the topic studied in the experiment.



Allan Rodrigues TE IT A-59

C. Conclusion.

In this experiment we learnt how to install and run typescript. We performed some simple programs in typescript and run it and got the desired output. We comprehended how to define variable, taking input from user, for loop etc in this experiment.

The significances of typescript are, It is more reliable & easier to refactor. This enables developers to evade errors & do reworks much easier. Typescript is more explicit, making types explicit focuses our attention on how exactly our system is built, and how different parts of it interact with each other.

D. References:

1. https://www.typescriptlang.org/assets/typescript-handbook.pdf
2. http://basarat.gitbooks.io/typescript/
3. https://robertcooper.me/post/get-started-with-typescript-in-2019