

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 – 3: To Perform Inheritance example using TypeScript.

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 basic inheritance features
 - To understand example of working with classes in TypeScript
 - To gain intuition not only for TypeScript's object-oriented language features, but for how and why use them.
3. **Outcomes:** After study of this experiment, the students will be able
 - To install Typescript.
 - Write code, compile and execute the code to achieve inheritance.
 - To know syntaxes for different types of inheritance
 - Learn how to use the inheritance to reuse the functionality of another class
4. **Prerequisite:** Basic knowledge of JavaScript is required, general concept of inheritance
5. **Requirements:** Personal Computer, Windows operating system, VSCode editor, browser, Internet Connection, google doc.
6. **Pre-Experiment Exercise:**
Brief Theory: Refer shared material
7. **Laboratory Exercise**
 - A. **Procedure:**
 - a. **Answer the following:**
 - Inheritance vs Composition(write in hand, scan and paste)

Allan Rodrigues TE ITA-59		Rajdhani DATE / /
Q.7 A	a) Inheritance vs Composition	
	Inheritance	Composition
	1) It does not allow code re-use	1) Composition allows code-reuse
	2) In inheritance, we will need to extend classes	2) In composition, we will no need to extend classes.
1	3) Inheritance is less flexible compared to composition	3) The composition is more flexible
	4) In inheritance, mixin plays a major role	4) In composition, we will no need to mixin

b. Attach screenshots:

- Typescript Program code and output with your own comments and indentation.

```

Go Run Terminal Help f2.ts - Typescript - Visual Studio Code
TS first.ts TS f2.ts 3 X f3.js TS loops.ts loops.js
TS f2.ts > s1 > read
1 class student {
2     name: string;
3
4     constructor(name: string) {
5         this.name = name;
6     }
7     submit(): void {
8         console.log("does submission");
9     }
10 }
11 class s1 extends student {
12
13     read(): void {
14         console.log(`does reading: ${this.name}`)
15     }
16 }
17
18 let s2 = new s1("Allan");
19 s2.read();
20 s2.submit();

```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP\Desktop\Allan\sem-VI\WL\Typescript>tsc f2.ts

C:\Users\HP\Desktop\Allan\sem-VI\WL\Typescript>node f2.js
does reading: Allan
does submission

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

```
Go Run Terminal Help f3.ts - Typescript - Visual Studio Code
TS first.ts x TS f2.ts 3 pbjs TS f3.ts x TS pb.ts 6
TS f3.ts > bird > constructor
1 class animal {
2     name: string;
3     constructor(name: string) {
4         this.name = name;
5     }
6     sleep(): void {
7         console.log("sleeping")
8     }
9 }
10 class bird extends animal {
11
12     constructor(name: string) {
13         console.log("sleeping1")
14         super(name)
15     }
16 }
17 class hen extends bird {
18     constructor(name: string) {
19         console.log("sleeping2")
20         super(name)
21     }
22 }
23 let heena = new hen("sfit");
24 // heena.eat();
25 // heena.fly();
26 // heena.sleep();
```

```
C:\Windows\System32\cmd.exe

C:\Users\HP\Desktop\Allan\sem-VI\WL\Typescript>tsc f3.ts

C:\Users\HP\Desktop\Allan\sem-VI\WL\Typescript>node f3.js
sleeping2
sleeping1

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

```
Go Run Terminal Help pb.ts - Typescript - Visual Studio Code
TS first.ts TS f2.ts 3 pb.js TS f3.ts TS pb.ts 6 X
TS pb.ts > office > constructor
1 class staff {
2     name: string;
3     constructor(name: string) {
4         this.name = name;
5     }
6     code(): void {
7         console.log("Coding:" + this.name)
8     }
9 }
10 class teacher extends staff {
11     subject(): void {
12         console.log("Teaching")
13     }
14 }
15 class office extends teacher {
16     constructor(name: string) {
17         super(name)
18         console.log(`My name is ${name}`)
19     }
20     grade(): void {
21         console.log("office")
22     }
23 }
24 let example = new office("Allan");
25 example.code();
26 example.subject();
27 example.grade();
28
29
```

```
C:\Windows\System32\cmd.exe
C:\Users\HP\Desktop\Allan\sem-VI\WL\Typescript>tsc pb.ts
C:\Users\HP\Desktop\Allan\sem-VI\WL\Typescript>node pb.js
My name is Allan
Coding:Allan
Teaching
office
C:\Users\HP\Desktop\Allan\sem-VI\WL\Typescript>
```

8. Post-Experiments Exercise

A. Extended Theory:

Nil

B. Questions:

- What are Interfaces in TypeScript? (write in hand, scan and paste)

Q8

B



What are interfaces in typescript.

Interface is a structure that defines the contract in your application. It defines the syntax for classes to follow. Classes that are derived from an interface must follow the structure provided by their interface. The typescript compiler does not convert interface to javascript. It uses interface for type checking. This is also known as "duck typing" or "Structural subtyping".

An interface is defined with the keyword `interface` & it can include properties & method declarations using a function or an arrow fun.

CS Scanned with CamScanner

- open up `src/index.ts` in your code editor. Enter the following code:

```
let a = 1 + 2
let b = a + 3
let c = {
  apple: a,
  banana: b
}
let d = c.apple * 4
```

1. hover over `a`, `b`, `c`, and `d`, and notice how and what TypeScript infers the types of all your variables
2. Play around with your code a bit. See if you can:
 - a. Get TypeScript to show a red squiggly when you do something invalid (we call this "throwing a `TypeError`").
 - b. Read the `TypeError`, and try to understand what it means. (execute code, attach ss)

```
Go Run Terminal Help index.ts - Exp3 -

TS index.ts X

TS index.ts > [c] c > banana

1 //Allan
2 let a = 1 + 2
3 let b = a + 3
4 let c = {
5     apple: a,
6     banana: b
7 }
8 let d = c.apple * 4
9
```

```
Go Run Terminal Help index.ts -

TS index.ts 2 X

TS index.ts > ...

1 //Allan
2 let a = 1 + 2
3 let b = a + 3
4 let c = {
5     apple: a,
6     banana: b
7 }
8 let b = c.apple * 4
9
```



```
Go Run Terminal Help index.ts - Exp3 - Visual Studio Code
TS index.ts 2 X
TS index.ts > ...
1 //Allan
2 let a = 1 + 2
3 let b = a + 3
4 let c = {
5     //
6 }
7
8 let b = c.apple * 4
9
```

Cannot redeclare block-scoped variable 'b'. ts(2451)

[View Problem](#) No quick fixes available

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.

Rajdhani
DATE / /

Q.8
c) Conclusion.

In this experiment we learnt features of inheritance. we comprehended working with classes in typescript. The goal of typescript is to help catch mistakes early through a type system & to make javascript development more efficient.

D. References:

1. <https://www.typescriptlang.org/assets/typescript-handbook.pdf>
2. <http://basarat.gitbooks.io/typescript/>
3. <https://www.tutorialsteacher.com/typescript/typescript-interface>
4. Programming TypeScript Making Your JavaScript Applications Scale, by Boris Cherny