DOP: / /2023 **DOS: / /2023**

Experiment No:2

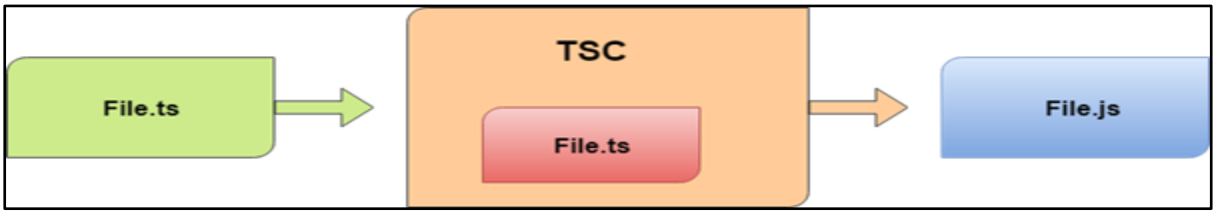
Title: 2.1 Installation of Typescript in windows, 2.2 Small code snippets for programs like Hello World, 2.3. Access Modifiers example using TypeScript,2.4 Inheritance example using TypeScript.

2.1Installation of Typescript in windows.

Theory:

- **Typescript:**

TypeScript is an open-source pure object-oriented programming language. It is a strongly typed superset of JavaScript which compiles to plain JavaScript. It contains all elements of the JavaScript. It is a language designed for large-scale JavaScript application development, which can be executed on any browser, any Host, and any Operating System. The TypeScript is a language as well as a set of tools. TypeScript is the ES6 version of JavaScript with some additional features.



TypeScript because of the following **benefits**.

- TypeScript supports Static typing, Strongly type, Modules, Optional Parameters, etc.
- TypeScript supports object-oriented programming features such as classes, interfaces, inheritance, generics, etc.
- TypeScript is fast, simple, and most importantly, easy to learn.
- TypeScript provides the error-checking feature at compilation time.
- TypeScript supports all JavaScript libraries because it is the superset of JavaScript.
- TypeScript support reusability because of the inheritance.
- TypeScript supports the latest JavaScript features, including ECMAScript 2015.
- TypeScript gives all the benefits of ES6 plus more productivity.
- Developers can save a lot of time with TypeScript.

🔷TypeScript Installation:

Pre-requisite to install TypeScript

- Text Editor or IDE
- Node.js Package Manager (npm)
- The TypeScript compiler

Step-1: Install Node.js. It is used to setup TypeScript on our local computer.

To install Node.js on Windows, go to the following link https://nodejs.org/en/download/

Step-2: To verify the installation was successful, enter the following command in the Terminal Window.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\priyush>node -v
v18.13.0

C:\Users\priyush>npm -v
8.19.3
```

Step-3 Install TypeScript. To install TypeScript, enter the following command in the Terminal Window.

```
$ npm install typescript --save-dev        //As dev dependency

$ npm install typescript -g                //Install as a global module

$ npm install typescript@latest -g         //Install latest if you have an older version
```

Step-4 To verify the installation was successful, enter the command $ tsc -v in the Terminal Window.

```
C:\Users\priyush>npm install -g typescript

changed 1 package, and audited 2 packages in 13s

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 8.19.3 -> 9.3.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.3.1
npm notice Run npm install -g npm@9.3.1 to update!
npm notice

C:\Users\priyush>tsc -v
Version 4.9.4

C:\Users\priyush>
```

2.2

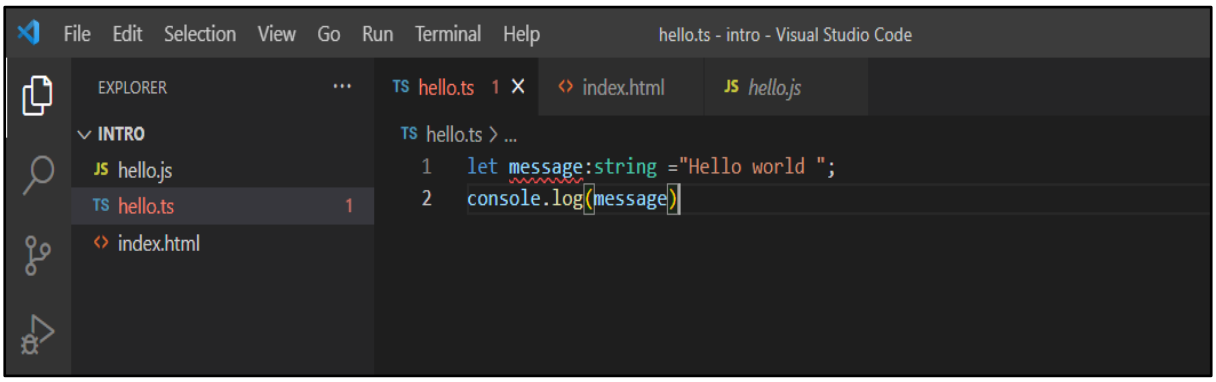
Title: Hello World Program in Typescript.

Theory:

We are going to learn how we can write a program in TypeScript, how to compile it, and how to run it. Also, we will see how to compiles the program and shows the error, if any.

Let us write a program in the text editor, save it, compile it, run it, and display the output to the console. To do this, we need to perform the following steps.

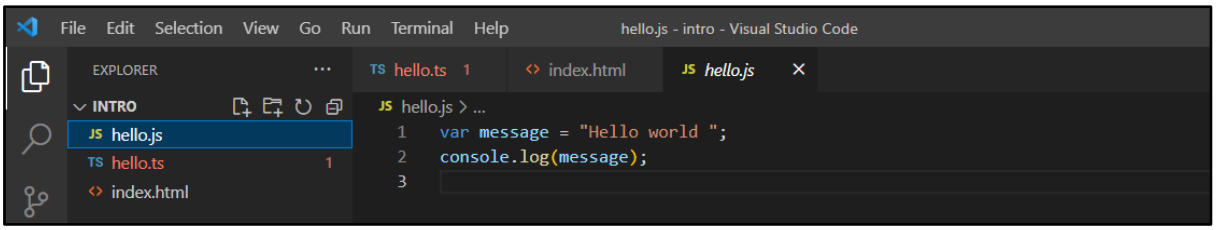
Step-1 Open the Text Editor and write/copy the following code.



Step-2 Save the above file as "**.ts**" extension.

Step-3 Compile the TypeScript code. To compile the source code, open the **command prompt**, and then goes to the file directory location where we saved the above file. For example, if we save the file on the desktop, go to the terminal window and type: - **cd Desktop/folder name**. Now, type the following command tsc **fileName. Ts** for compilation and press **Enter**.

It will generate JavaScript file with ".js" extension at the same location where the TypeScript source file exists. The below ".js" file is the output of TypeScript (.ts) file.



Step-4 Now, to run the above JavaScript file, type the following command in the terminal window: node filename.js and press Enter. It gives us the final output as:

```
PS C:\Users\priyush\Desktop\Typescript\intro> tsc hello.ts
PS C:\Users\priyush\Desktop\Typescript\intro> tsc hello.ts
PS C:\Users\priyush\Desktop\Typescript\intro> tsc hello.ts
PS C:\Users\priyush\Desktop\Typescript\intro> node hello.js
Hello world
PS C:\Users\priyush\Desktop\Typescript\intro>
```

2.3

Title: Access Modifiers example using TypeScript.

Theory:

- **Access Modifiers:**

Typescript allows us to use access modifiers at the class level. It gives direct access control to the class member. These class members are functions and properties. We can use class members inside its own class, anywhere outside the class, or within its child or derived class.

The access modifier increases the security of the class members and prevents them from invalid use. We can also use it to control the visibility of data members of a class. If the class does not have to be set any access modifier, TypeScript automatically sets public access modifier to all class members.

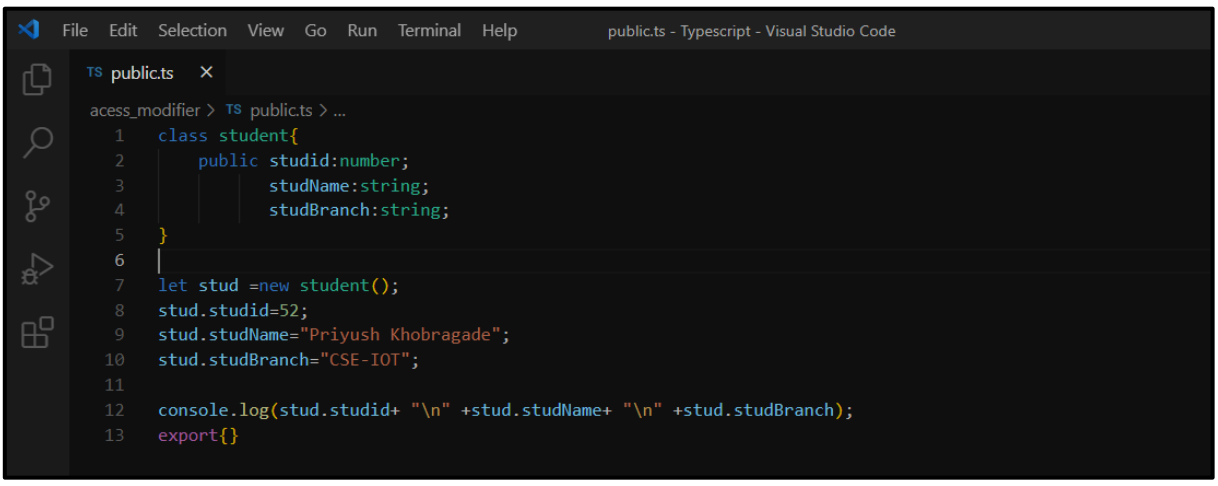
The TypeScript access modifiers are of three types. These are:

1. Public
2. Private
3. Protected.

1.Public

In TypeScript by default, all the members (properties and methods) of a class are public. So, there is no need to prefix members with this keyword. We can access this data member anywhere without any restriction.

Input:



Output:

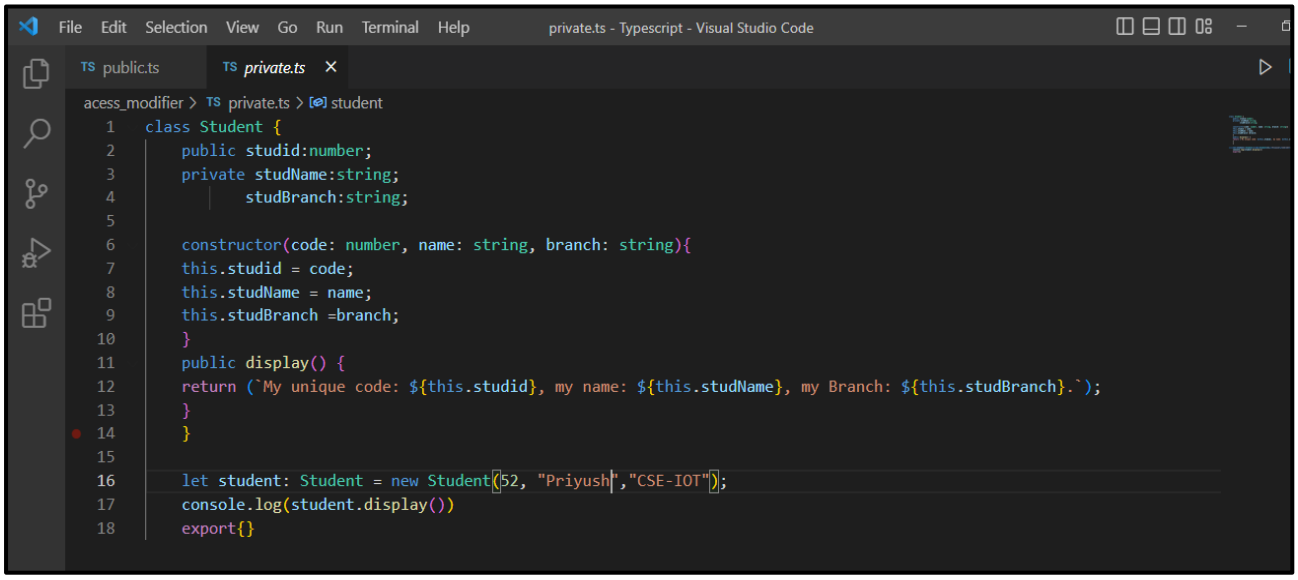
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\priyush\Desktop\Typescript> cd acess_modifier
PS C:\Users\priyush\Desktop\Typescript\acess_modifier> tsc public.ts
PS C:\Users\priyush\Desktop\Typescript\acess_modifier> node  public.js
52
Priyush Khobragade
CSE-IOT
PS C:\Users\priyush\Desktop\Typescript\acess_modifier>
```

Private

The private access modifier cannot be accessible outside of its containing class. It ensures that the class members are visible only to that class in which it is containing.

Input:



Output:

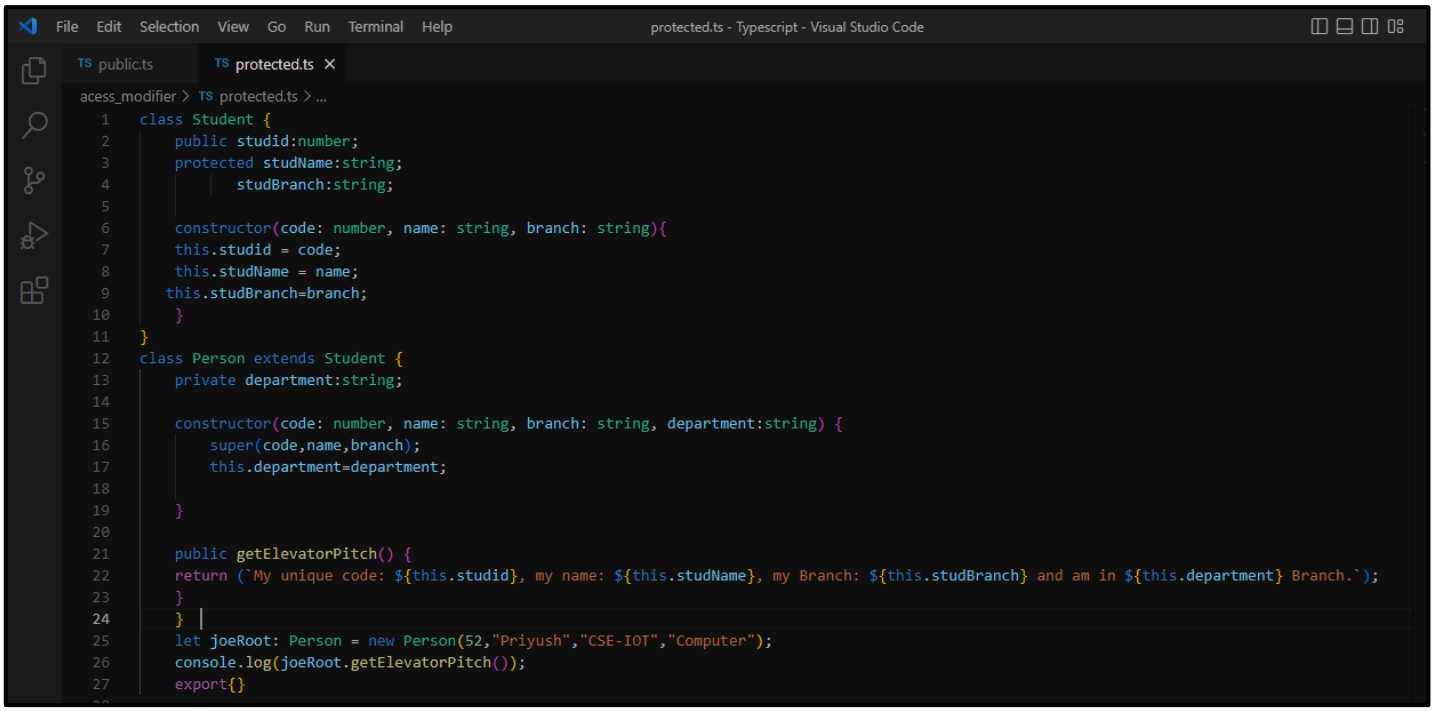
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\priyush\Desktop\Typescript\acess_modifier> tsc private.ts
PS C:\Users\priyush\Desktop\Typescript\acess_modifier> node private.js
My unique code: 52, my name: Priyush, my Branch: CSE-IOT.
PS C:\Users\priyush\Desktop\Typescript\acess_modifier>
```

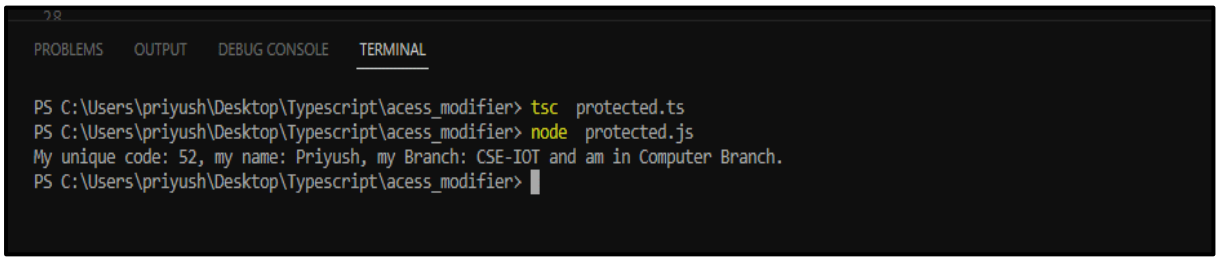
Protected:

A Protected access modifier can be accessed only within the class and its subclass. We cannot access it from the outside of a class in which it is containing.

Input:



Output:



2.4

Title: Inheritance example using TypeScript.

Theory:

- **TypeScript Inheritance:**

Inheritance is an aspect of OOPs languages, which provides the ability of a program to **create a new class from an existing class.** It is a mechanism which acquires the **properties** and **behaviours** of a class from another class. The class whose members are inherited is called the **base class**, and the class that inherits those members is called the **derived/child/subclass.** In child class, we can override or modify the behaviours of its parent class.

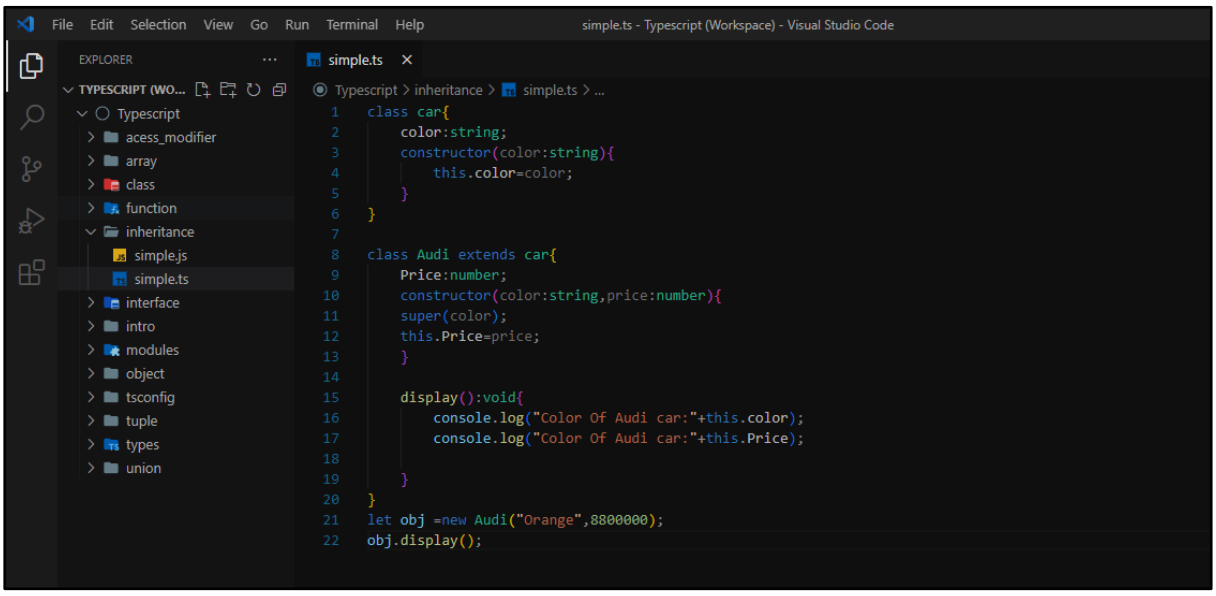
The TypeScript uses class inheritance through the extends keyword. TypeScript supports only **single** inheritance and **multilevel** inheritance. It doesn't support multiple and hybrid inheritance.

Syntax

1. We can declare a class inheritance as below.

```
1. class sub_class_name extends super_class_
2. {
3.         // methods and fields
4. {
```

Simple Inheritance:



Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\priyush\Desktop\Typescript> cd inheritance
PS C:\Users\priyush\Desktop\Typescript\inheritance> tsc simple.ts
PS C:\Users\priyush\Desktop\Typescript\inheritance> node  simple.js
Color Of Audi car:Orange
Color Of Audi car:8800000
PS C:\Users\priyush\Desktop\Typescript\inheritance>
```

- **Types of Inheritance:**

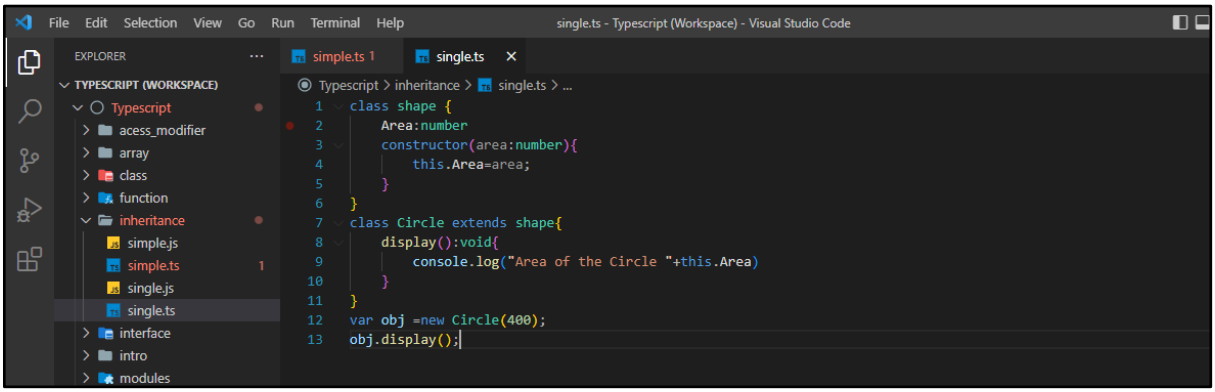
We can classify the inheritance into the five types. These are:

1. Single Inheritance
2. Multilevel Inheritance
3. Multiple Inheritance
4. Hierarchical Inheritance
5. Hybrid Inheritance

Single Inheritance

Single inheritance can inherit properties and behaviour from at most one parent class. It allows a derived/subclass to inherit the properties and behaviour of a base class that enable the code reusability as well as we can add new features to the existing code.

Input:



Output:

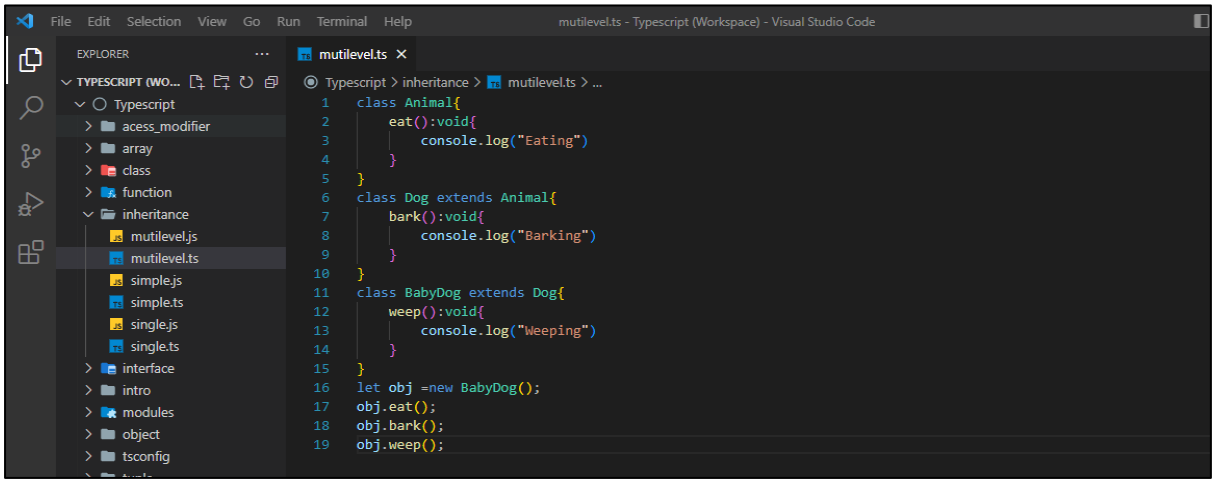
```
PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\priyush\Desktop\Typescript\inheritance> tsc single.ts
PS C:\Users\priyush\Desktop\Typescript\inheritance> node  single.js
Area of the Circle 400
PS C:\Users\priyush\Desktop\Typescript\inheritance>
```

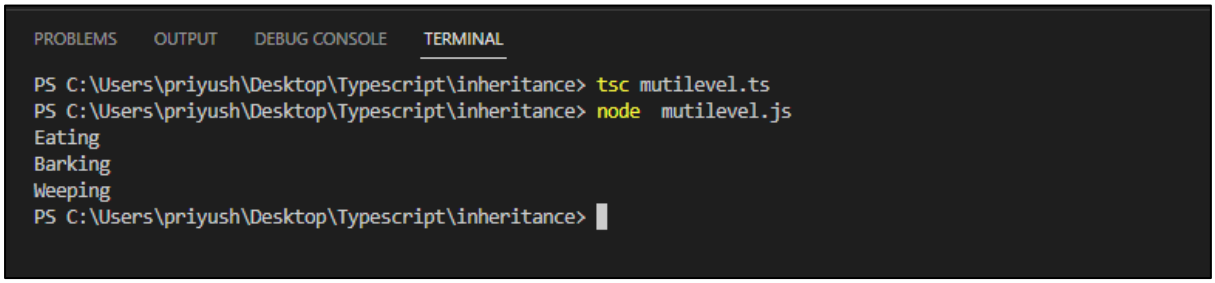
Multilevel Inheritance

When a derived class is derived from another derived class, then this type of inheritance is known as multilevel inheritance. Thus, a multilevel inheritance has more than one parent class.

Input:



Output:



Multiple Inheritance:

When an object or class inherits the characteristics and features form more than one parent class, then this type of inheritance is known as multiple inheritance.

Hierarchical Inheritance:

When more than one subclass is inherited from a single base class, then this type of inheritance is known as hierarchical inheritance.

Hybrid Inheritance:

When a class inherits the characteristics and features from more than one form of inheritance, then this type of inheritance is known as Hybrid inheritance.

Conclusion: - We Successfully implement Installation of Typescript in windows, Small code snippets for programs like Hello World, Access Modifiers example using TypeScript, Inheritance example using TypeScript.