Aim-

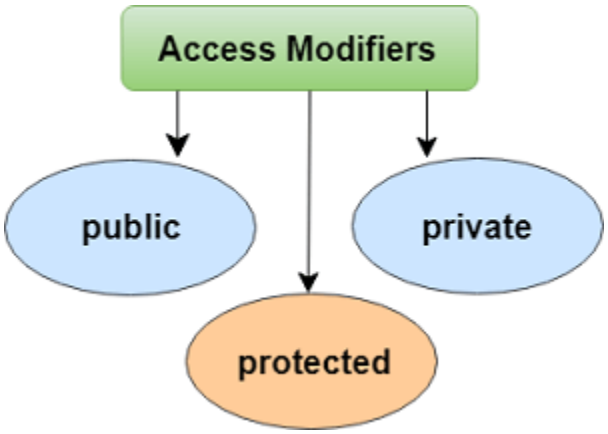
1. Small code snippets for programs like Hello World, and Calculator using TypeScript.
2. Inheritance example using TypeScript.
3. Access Modifiers example using TypeScript.

Theory-

TypeScript Access Modifiers
Like other programming languages, 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 their 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 the public access modifier to all class members.

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

1. Public
2. Private
3. Protected.



Output-

1. Small code snippets for programs like Hello World, and Calculator using TypeScript.

```
console.log("Hello World!!");
```

```
E:\Mikil\TE\Sem 6\Labs\Lab Web X\Exp 2>tsc hello.ts

E:\Mikil\TE\Sem 6\Labs\Lab Web X\Exp 2>node hello.js
Hello World!!

E:\Mikil\TE\Sem 6\Labs\Lab Web X\Exp 2>
```

```
var a:number = 20;
var b:number = 10;

function add(a:number, b:number):number{
return a+b;
}
function sub(a:number, b:number):number{
return a-b;
}
function div(a:number, b:number):number{
return a/b;
}
function mul(a:number, b:number):number{
return a*b;
}

console.log(add(a,b));
console.log(sub(a,b));
console.log(div(a,b));
console.log(mul(a,b));
```

```
E:\Mikil\TE\Sem 6\Labs\Lab Web X\Exp 2>tsc calculator.ts

E:\Mikil\TE\Sem 6\Labs\Lab Web X\Exp 2>node calculator.js
30
10
2
200

E:\Mikil\TE\Sem 6\Labs\Lab Web X\Exp 2>
```

2. Inheritance example using TypeScript.

```
class Person {
```

```
firstName: string;
lastName: string;
constructor(firstName, lastName) {
this.firstName = firstName;
this.lastName = lastName;
}
}
class Employee extends Person {
employeeID: number;
constructor(firstName, lastName, employeID) {
super(firstName, lastName);
this.employeeID = employeID;
}
}

class Manager extends Employee{
department :string;
constructor(firstName, lastName, employeID, department) {
super(firstName, lastName, employeID);
this.department = department;
}
}
let manager = new Manager('Mehul','Sharma', 123, 'Sales');
console.log("Name of manager: "+manager.firstName + manager.lastName);
console.log("Emplpoyee ID: "+ manager.employeeID);
console.log("Manager's Department: "+manager.department);
```

```
E:\Mikil\TE\Sem 6\Labs\Lab Web X\Exp 2>tsc inheritance.ts

E:\Mikil\TE\Sem 6\Labs\Lab Web X\Exp 2>node inheritance.js
Name of manager: MehulSharma
Emplpoyee ID: 123
Manager's Department: Sales

E:\Mikil\TE\Sem 6\Labs\Lab Web X\Exp 2>
```

3. Access Modifiers example using TypeScript.

```
class Student{
public name :string = "Mikil";
protected roll :number = 37;
private accNo :number = 12345;
getAccNo()
```

```
{
return this.accNo;
}
}

class Person extends Student{
constructor()
{
super();
}
display()
{
return "Roll number of student: "+this.roll;
}
}

let s :Student = new Student();
let p :Person = new Person();

console.log("Name of student: "+s.name);
console.log(p.display());
console.log("Account number of student: "+s.getAccNo());
```

```
E:\Mikil\TE\Sem 6\Labs\Lab Web X\Exp 2>tsc accessModifier.ts

E:\Mikil\TE\Sem 6\Labs\Lab Web X\Exp 2>node accessModifier.js
Name of student: Mikil
Roll number of student: 37
Account number of student: 12345

E:\Mikil\TE\Sem 6\Labs\Lab Web X\Exp 2>
```

Conclusion-

Thus we successfully learned the basics of typescript and performed various experiments. We implemented different programs in typescript to understand the programming language.