PROGRAM-7:

Aim:Write a typescript program to work with different types of variables ,functions and run the programs using node environment.

Installation of TypeScript:

Step-1:

Install Node.js. It is used to setup TypeScript on our local computer and verify the installation is done or not by the command

node -v

```
C:\20481A05C2>node -v
v18.14.2
```

Step2:

Install TypeScript,we use the following commands

npm install typescript - -save-dev

npm install typescript -g

```
C:\20481A05C2>npm install typescript --save-dev

up to date, audited 82 packages in 922ms

8 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
C:\20481A05C2>npm install typescript -g

changed 1 package in 950ms

C:\20481A05C2>tsc -v
Version 5.0.3
```

Example Program:

exp7.ts

```
function student(name:String,branch:String,cgpa:number)
{
   console.log("student name:"+name)
   console.log("student branch:"+branch)
   console.log("student cgpa:"+cgpa)
}
function studentdefault(name:string="ravi",branch:string="cse",cgpa:number=95)
{
   console.log("student name:"+name)
   console.log("student branch:"+branch)
   console.log("student cgpa:"+cgpa)
}
function studentdetails(name:string,rollno:string,sgpa:number)
{
   console.log("student name:"+name)
   console.log("student rollno:"+rollno)
   console.log("student sgpa:"+sgpa)
}
function studentmarks(...s:number[])
{
  var i;
  var sum:number=0;
  var avg;
  for(i=0;i<s.length;i++)
  {
   sum=sum+s[i];
  }
  console.log("total marks:"+sum)
  console.log("Average marks:"+sum/(s.length))
}
student("dinesh","cse",89)
studentdefault()
studentdetails("vamsi","cse",85)
studentmarks(94,97,99,66,89,85)
```

```
studentdetails("sandeep","512",80)
studentmarks(100, 67, 83, 76, 98);
```

Example Program:

Exp7.js

```
function student(name, branch, cgpa) {
    console.log("student name:" + name);
    console.log("student branch:" + branch);
    console.log("student cgpa:" + cgpa);
}
function studentdefault(name, branch, cgpa) {
    if (name === void 0) { name = "ravi"; }
    if (branch === void 0) { branch = "cse"; }
    if (cgpa === void 0) { cgpa = 95; }
    console.log("student name:" + name);
    console.log("student branch:" + branch);
    console.log("student cgpa:" + cgpa);
}
function studentdetails(name, rollno, sgpa) {
    console.log("student name:" + name);
    console.log("student rollno:" + rollno);
    console.log("student sgpa:" + sgpa);
}
function studentmarks() {
    var s = [];
    for (var _i = 0; _i < arguments.length; _i++) {
```

```
        s[_i] = arguments[_i];
    }
    var i;
    var sum = 0;
    var avg;
    for (i = 0; i < s.length; i++) {
        sum = sum + s[i];
    }
    console.log("total marks:" + sum);
    console.log("Average marks:" + sum / (s.length));
}
student("dinesh", "cse", 89);
studentdefault();
studentdetails("vamsi", "cse", 85);
studentmarks(94, 97, 99, 66, 89, 85);
studentdetails("sandeep", "512", 80);
studentmarks(100, 67, 83, 76, 98);
```

output:

```
D:\dinesh>tsc exp7.ts
D:\dinesh>node exp7.js
student name:dinesh
student branch:cse
student cgpa:89
student name:ravi
student branch:cse
student cgpa:95
```

student name:vamsi

student rollno:cse

student sgpa:85

total marks:530

Average marks:88.33333333333333

student name:sandeep

student rollno:512

student sgpa:80

total marks:424

Average marks:84.8

PROGRAM:8

Aim:Write a typescript program to work with classes.

CLASSES IN TYPESCRIPT:

In object-oriented programming languages like Java, classes are the fundamental entities which are used to create **reusable** components. It is a group of objects which have common properties. In terms of OOPs, a class is a **template** or **blueprint** for creating objects. It is a logical entity.

A class definition can contain the following properties:

- **Fields:** It is a variable declared in a class.
- **Methods:** It represents an action for the object.
- **Constructors:** It is responsible for initializing the object in memory.
- **Nested class and interface:** It means a class can contain another class.

TypeScript is an Object-Oriented JavaScript language, so it supports object-oriented programming features like classes, interfaces, polymorphism, data-binding, etc. JavaScript **ES5** or **earlier version** did not support classes. TypeScript support this feature from **ES6** and **later version**. TypeScript has **built-in** support for using classes because it is based on ES6 version of JavaSript. Today, many developers use class-based object-oriented programming languages and compile them into JavaScript, which works across all major browsers and platforms.

Syntax to declare a class:

```
class <class_name>{
   field;
   method;
}
```

The TypeScript compiler converts class into JavaScript code.

Creating an object of class:

A class creates an object by using the **new** keyword followed by the **class name**. The new keyword allocates memory for object creation at runtime. All objects get memory in heap memory area. We can create an object as below.

Syntax:

let object_name = new class_name(parameter)

1. new keyword: it is used for instantiating the object in memory.
2. The right side of the expression invokes the constructor, which can pass values.

Object Initialization:

Object initialization means storing of data into the object. There are three ways to initialize an object. These are:

1. By reference variable
2. By method
3. By constructor

PROGRAM:

exp8.ts

```
class Student
{
  studcode:number;
   studname:string;
   grade:string;
   constructor(code:number,name:string,grade:string){
      this.studname=name;
      this.studcode=code;
      this.grade=grade;
   }
   display():void{
      console.log("name:",this.studname);
      console.log("code:",this.studcode);
      console.log("grade:",this.grade);
   }
}
let obj1=new Student(9491825377,'dinesh','A+');
obj1.display();
```

exp8.js:

```
var Student = /** @class */ (function () {
   function Student(code, name, grade) {
```

```
        this.studname = name;
        this.studcode = code;
        this.grade = grade;
    }
    Student.prototype.display = function () {
        console.log("name:", this.studname);
        console.log("code:", this.studcode);
        console.log("grade:", this.grade);
    };
    return Student;
}());
var obj1 = new Student(9491825377, 'dinesh', 'A+');
obj1.display();
```

OUTPUT: