**S8\_\_ Class**

**Properties:** its nothing but natural extension of data fields. it’s a member of class provide flexible mechanism to expose private field. It also used for Encapsulation. Its special method called accessor. Its having two values Get: Return the property value. and Set: assign new value.

**Getters and Setter:-** Getter is used to return values but setter is used to set new value.

**Class: -** user defined blue print or prototype or Template. Its datatype. It represent entity or object. Define state and behaviour of an object. Using class we can create object. Class can be provide derived and child class. Class keyword is used to declare and define class body.its **refrence type.** Default access modifier is internal. Default access modifier for method and variable is private. It contain memebrs of element called class member.

**Object:** represents the real life entities. Each non static method is having object. New keyword is used to create instance or Object. When program runs object has been created. It’s a tangible entity. It contain identity, state, and behaviour.

**Constructor:** its special method of class which gets automatically invoked whenever instance of class is created. Constructor have same name as class. It cannot be abstract final and synchronised. It’s like method. In a class only one static constructor. No return type. Static constructor can’t be parameterised.

**This :-** this keyword refer the object. Used to called object or method.

**Program**

let names = ["Amit", "Anis", "Keshav"];

class personDetail {

    constructor(firstName, lastName, age, gender) {

        this.firstName = firstName;

        this.lastName = lastName;

        this.age = age;

        this.gender = gender;

    }

    get fullName() { //get

        return `${this.firstName} ${this.lastName} ${this.age}`

    }

    set fullName(fullName) {

        const [firstName, lastName] = fullName.split(" ");

        this.firstName = firstName;

        this.lastName = lastName;

    }

    addition() {

        var prompt = require('prompt-sync')();

        var number1 = Number(prompt("Enter Number: "));

        var number2 = Number(prompt("Enter Number: "));

        var number3 = number1 + number2;

        console.log("Addition is:" + number3);

    }

    static multiplication() {

        var prompt = require('prompt-sync')();

        var number1 = prompt("Enter Number: ");

        var number2 = prompt("Enter Number: ");

        var number3 = number1 \* number2;

        console.log("Multiplication is:" + number3);

    }

}

class functionone {

     constructor(firstName, lastName, age, gender) {

        this.firstName = firstName;

        this.lastName = lastName;

        this.age = age;

        this.gender = gender;

    }

    printingName() {

        console.log("Inside the anothr class");

        console.log(names);

    }

}

const persons = new personDetail("Imran", "Shaih", 22, 'M')

// console.log(persons);

// console.log(persons.fullName);

// persons.firstName = "Amit";

// persons.fullName = "Gujar"

// persons.fullName = "Amir Shaikh";

// console.log(persons.fullName);

console.log("\*\*\*Calling Methods\*\*\*\*")

// personDetail.multiplication();  ///calling static method

// persons.multiplication();

// persons.addition();

const printingNames=new functionone(this.printingName);

printingNames.printingName();