1. **https://github.com/rvsp/typescript-oops/blob/master/Practice/Movie.md**
2. **https://github.com/rvsp/typescript-oops/blob/master/Practice/class-circle.md**
3. **Write a “person” class to hold all the details.**
4. **write a class to calculate uber price.**

**//https://github.com/rvsp/typescript-oops/blob/master/Practice/Movie.md**

class Movie {

    constructor(title,studio,rating='PG') {

        this.title=title;

        this.studio=studio;

        this.rating=rating;

    }

    static getPG(arr) {

        //let data = new Movie[mov.length];

        let data=[]

        for (let i = 0; i < arr.length; i++) {

                if (arr[i].rating==='PG') {

                    data.push(arr[i]);

                }

        }

        return data;

    }

  }

  let mov1 = new Movie('Casino Royale','EonProductions','PG13');

  let mov2 = new Movie('Sivaji','Geministudios','P');

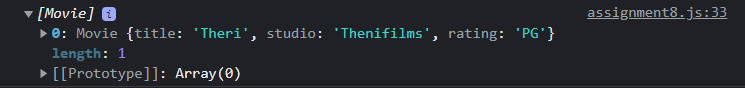
  let mov3 = new Movie('Billa','Sivajistudio','Q');

  let mov4 = new Movie('Theri','Thenifilms');

  let arr=[mov1,mov2,mov3,mov4]

  console.log(Movie.getPG(arr));

**Output:**

****

**//https://github.com/rvsp/typescript-oops/blob/master/Practice/class-circle.md**

class Circle {

constructor(radius = '1.0', color = 'red') {

this.radius = radius

this.color = color

}

getRadius(radius) {

return radius

}

setRadius() {

this.radius = radius;

}

getColour(color) {

return color

}

setColour() {

this.color = color;

}

toString(){

return (`Radius=${this.radius},Color=${this.color}`)

}

getArea(){

return((22/7)\*this.radius\*this.radius)

}

getCircumference(){

return(2\*(22/7)\*this.radius)

}

}

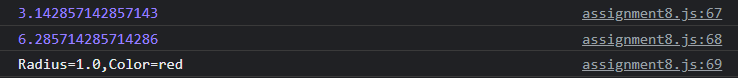
let circle=new Circle();

console.log(circle.getArea()) ;

console.log(circle.getCircumference()) ;

console.log(circle.toString()) ;

**Output:**



**//Write a “person” class to hold all the details.**

class Person {

    constructor(name,age,height,weight,email,phone) {

        this.name = name;

        this.age = age;

        this.height = height;

        this.weight = weight;

        this.email = email;

        this.phone = phone;

        console.log("I am " + this.name);

    }

    greetings() {

        console.log("Welcome to Method greeting");

    }

    getFood() {

        console.log("I am food method");

    }

    static displayWalk() {

         console.log("I am static method with name displaywalk");

        }

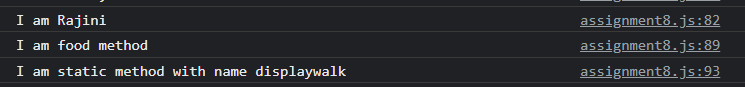
}

let person1 = new Person('Rajini',72,6.2,70,'rajini@hotmail.com',9183097);

person1.getFood();

Person.displayWalk();

**Output:**



**//write a class to calculate uber price.**

class Uber{

    constructor(basefare,timerate,distancerate,bookingfee,tollfee){

     this.basefare=basefare;

     this.timerate=timerate;

     this.distancerate=distancerate;

     this.bookingfee=bookingfee;

     this.tollfee=tollfee;

    }

    calculatePrice()

    {

      return(((this.basefare + this.timerate)\*this.distancerate)+this.bookingfee + this.tollfee )

    }

}

let uber=new Uber(50,2,20,30,20);

console.log(uber.calculatePrice())

**Output:**

