1. The class Movie is stated below. An instance of class Movie represents a film. This class has the following three properties:

* title, which is a String representing the title of the movie
* studio, which is a String representing the studio that made the movie
* rating, which is a String representing the rating of the movie (i.e. PG­13, R, etc)

a) Write a constructor for the class Movie, which takes a String representing the title of the movie, a String representing the studio, and a String representing the rating as its arguments, and sets the respective class properties to these values.

b) The constructor for the class Movie will set the class property rating to "PG" as default when no rating is provided.

c) Write a method getPG, which takes an array of base type Movie as its argument, and returns a new array of only those movies in the input array with a rating of "PG". You may assume the input array is full of Movie instances. The returned array need not be full.

d) Write a piece of code that creates an instance of the class Movie with the title “Casino Royale”, the studio “Eon Productions”, and the rating “PG­13”

**Solution:**

class Movie{

    constructor(title,studio,rating){

        this.title = title

        this.studio = studio || "PG"

        this.rating = rating

    }

}

// A piece of code that creates an instance of the class Movie.

const CasinoRoyal = new Movie("Casino Royal","Eon Productions","PG13");

1. **Circle – Class:**

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

**Solution:**

class Circle{

    constructor(radius,color){

        this.\_radius =radius;

        this.\_color = color;

    }

    // +Circle(radius:double)

     get double(){

        return this.\_radius \* 2;

    }

    // +Circle(radius:double, color:String)

      color(){

        this.color;

    }

    // getRadius Double

    get getRadius(){

        return this.\_radius \* 2;

    }

    // setRadius

    set getRadius(input){

        if(input == "void"){

            console.log("Void")

        }

        else{

            console.log("Error: Your Input is not Valid")

        }

    }

    // get Color

    get Color(){

        return this.\_color;

    }

    // set Color

    set Color(input){

        if(input == "Blue"){

            console.log("Blue")

        }

        else{

            console.log("Error: Your Input is not Valid")

        }

    }

    // toString

    get toString(){

        return `Circle[radius=${this.\_radius}, color = ${this.\_color}]`

    }

    // getArea

    get getArea(){

        return Math.PI \* this.\_radius \* this.\_radius;

    }

    // get Circumference

    get getCircumference(){

        return 2 \* Math.PI \* this.\_radius;

    }

}

const obj = new Circle(1.0,"red");

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

class person{

    constructor(name,age,position,city){

        this.name = name;

        this.age = age;

        this.position = position;

        this.city = city

    }

}

var person1 = new person("SRIRAM GOKUL",30,"Sr Engineer","Chennai")

var person2 = new person("RAM",26,"Jr Engineer","Mumbai")

var person3 = new person("Deepika",45,"HR","New Delhi")

var person4 = new person("Athul",56,"CEO","Bangalore")

1. write a class to calculate the uber price.

class Uber{

    constructor(Basefare,costPerMinute,costPerKm,bookingFee){

        this.Basefare = Basefare;

        this.costPerMinute = costPerMinute;

        this.costPerKm = costPerKm;

        this.bookingFee = bookingFee;

    }

    // Price for the ride

    calculatePrice(time,distance){

       const totalTimeTravel = time \* this.costPerMinute;

       const totalDistance = distance \* this.costPerKm;

        const FinalPrice = this.Basefare + totalTimeTravel + totalDistance + this.bookingFee;

        return FinalPrice;

    }

}

const obj = new Uber(30,10,25,50);

const time = 80;

const distance = 50;

const UberPrice = obj.calculatePrice(time,distance);

console.log("Final Uber Ride Price is "  + UberPrice.toFixed(2));