**Class - Movie**

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='PG'){

        this.title=title;

        this.studio=studio;

        this.rating=rating;

    }

}

var movie1=new movie('Casino Royale','Eon Productions','PG­13');

console.log(movie1);

**Output:**

Movie

rating: "PG­13"

studio: "Eon Productions"

title: "Casino Royale

# 2) Circle - Class

Convert the UML diagram to Typescript class. - use number for double

class circle{

    constructor(radius,color){

        this.radius=radius;

        this.color=color;

    }

    circle(){

        console.log("this is circle")

    }

    circle(radius){

        console.log("method overriding")

    }

    circle(radius,color){

        console.log(this.radius,this.color+'final method overriding which accepts by javascripot')

    }

    getRadius(){

        console.log("the radius of the circle is "+ this.radius)

    }

    setRadius(newradius){

       this.radius=newradius;

    }

    getColor(){

        console.log('the color of the circle is '+ this.color)

    }

    setColor(newcolor){

        this.color=newcolor;

    }

    getArea(){

        console.log('the area of circle is '+ (Math.PI\*Math.pow(this.radius,2)))

    }

    getCircumference(){

        console.log('the circumference of a circle is '+ (2\*Math.PI\*this.radius))

    }

}

let circle1=new circle(1,"red");

circle1.getRadius();

circle1.setRadius(2)

circle1.getRadius();

circle1.getColor();

circle1.setColor("blue");

circle1.getColor()

circle1.getArea()

circle1.getCircumference()

**Output:**

the radius of the circle is 1

function.js:17 the radius of the circle is 2

function.js:24 the color of the circle is red

function.js:24 the color of the circle is blue

function.js:31 the area of circle is 12.566370614359172

function.js:34 the circumference of a circle is 12.566370614359172

**3)Write a “person” class to hold all the details:**

class person{

    constructor(name,age,profession){

        this.name=name;

        this.age=age;

        this.profession=profession;

    }

   getname(){

       console.log("name is "+ this.name)

   }

   getage(){

       console.log("age is "+ this.age);

   }

   getprofession(){

       console.log("profession is "+ this.profession)

   }

   setname(newname){

       this.name = newname;

   }

   setage(newage){

       this.age = newage

   }

   setprofession(newprofession){

       this.profession = newprofession

   }

}

var person1=new person();

person1.setname("Jeya");

person1.setage(30);

person1.setprofession("developer");

person1.getname();

person1.getage();

person1.getprofession();

**Output:**

name is Jeya

function.js:11 age is 30

function.js:15 profession is developer

**4)write a class to calculate uber price:**

class UberPriceCalculator{

    constructor(basefare,CPM,TR,CRD,SBM,ridedistance,bookingfee){

        this.basefare=basefare;

        this.CPM=CPM;

        this.TR=TR;

        this.CRD=CRD;

        this.SBM=SBM;

        this.ridedistance=ridedistance

        this.bookingfee=bookingfee;

    }

    getPrice(){

        console.log("THE RIDE PRICE IS "+(this.basefare + ((this.CPM \* this.TR) + (this.CRD \* this.ridedistance) \*this.SBM) + this.bookingfee ))

    }

     setPrice(newbasefare,newCPM,newTR,newCRD,newSBM,newridedistance,newbookingfee){

            this.basefare=newbasefare;

            this.CPM=newCPM;

            this.TR=newTR;

            this.CRD=newCRD;

            this.SBM=newSBM;

            this.ridedistance=newridedistance

            this.bookingfee=newbookingfee;

        }

}

var cost=new UberPriceCalculator();

cost.setPrice(100,5,20,15,10,3,20);

cost.getPrice();

**Output:**

THE RIDE PRICE IS 670