**OOPs In JS**

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”**

class movies

{

constructor(title, studio, rating = "PG")

{

this.title = title;

this.studio = studio;

this.rating = rating;

}

}

var moview1 = new movies ("Casino Royale","Eon Productions","PG.13");

var moview2 = new movies ("Casino Royale","Eon Productions");

console.log(moview1.title);

console.log(moview1.studio);

console.log(moview1.rating);

console.log(moview2.rating);

**Output:**

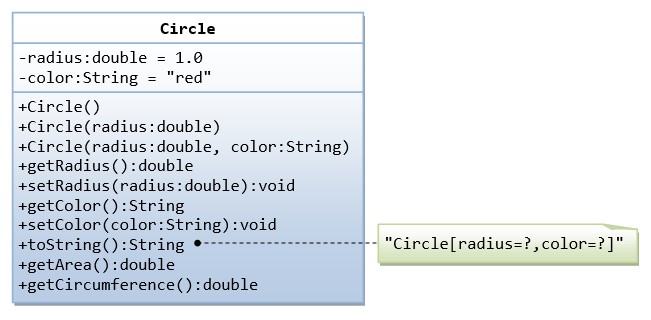
**Casino Royale**

**Eon Productions**

**PG.13**

**PG**

1. [**https://github.com/rvsp/typescript-oops/blob/master/Practice/class-circle.md**](https://github.com/rvsp/typescript-oops/blob/master/Practice/class-circle.md)



class circule {

constructor(r, c)

{

this.radius = r;

this.color = c;

}

get radiusCirucle()

{

return this.radius;

}

set radiusCirucle (radius)

{

this.radius = radius;

}

get colorCirucle ()

{

return this.color;

}

set colorCirucle(color)

{

this.color = color;

}

get toString()

{

return `"Cirucle [radius = ${this.radius}, color = ${this.color}]"`;

}

get areaCircule()

{

return Math.PI \* Math.pow(this.radius,2);

}

get cirumference ()

{

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

}

}

var foo = new circule (1.0,"Red");

console.log(foo.radiusCirucle);

foo.radiusCirucle = 3.1;

console.log(foo.radiusCirucle);

console.log(foo.colorCirucle);

foo.colorCirucle = "Yellow";

console.log(foo.colorCirucle);

console.log(foo.toString);

console.log(foo.areaCircule);

console.log(foo.cirumference);

**Output:**

**1**

**3.1**

**Red**

**Yellow**

**"Cirucle [radius = 3.1, color = Yellow]"**

**30.190705400997917**

**19.477874452256717**

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

class person {

constructor(Fn,Ln,Dob,add)

{

this.first = Fn;

this.last = Ln;

this.dateofBirth = Dob;

this.address = add;

}

}

var person1 = new person("Dinesh","Kumar","01-01-2023","Salem");

var person2 = new person("Ramesh","nandha","02-02-2023","Erode");

var person3 = new person("John","khan","03-03-2023","Hosur");

var person4 = new person("Vinesh","subu","04-04-2023","Karur");

console.log(person1.first);

console.log(person1.last);

console.log(person1.dateofBirth);

console.log(person1.address);

console.log(person2.first);

console.log(person2.last);

console.log(person2.dateofBirth);

console.log(person2.address);

console.log(person3.first);

console.log(person3.last);

console.log(person3.dateofBirth);

console.log(person3.address);

console.log(person4.first);

console.log(person4.last);

console.log(person4.dateofBirth);

console.log(person4.address);

console.log(`Over all person names are,

Person1 : ${person1.first}

Person2 : ${person2.first}

Person3 : ${person3.first}

Person4 : ${person4.first}`);

**Output:**

**Dinesh**

**Kumar**

**01-01-2023**

**Salem**

**Ramesh**

**nandha**

**02-02-2023**

**Erode**

**John**

**khan**

**03-03-2023**

**Hosur**

**Vinesh**

**subu**

**04-04-2023**

**Karur**

**Over all person names are,**

**Person1 : Dinesh**

**Person2 : Ramesh**

**Person3 : John**

**Person4 : Vinesh**

1. **Write a class to calculate the uber price.**

class uberprice {

constructor(bf,cpk,cpm,bookf)

{

this.barefare = bf;

this.costperkm = cpk;

this.costpermin = cpm;

this.bookingfee = bookf;

}

calculatePrice (distanceInKm,timeInMin)

{

var distanceCost = this.costperkm\*distanceInKm;

var timecost = this.costpermin\*timeInMin;

var totalPrice = this.barefare+distanceCost+timecost+this.bookingfee;

return totalPrice;

}

}

var calculator = new uberprice(60,15,2.5,30);

var distanceInKm = 6.0;

var timeInMin = 10;

var estimate = calculator.calculatePrice(distanceInKm,timeInMin);

console.log( `Uberprice : ${estimate}`);

**Output: Uberprice : 205**