**JAVASCRIPT DAY-8**

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

class person {

constructor(name,age,degree,email,address,mobilenumber)

{

this.name=name;

this.age=age;

this.degree=degree;

this.email=email;

this.address=address;

this.mobilenumber=mobilenumber;

}

}

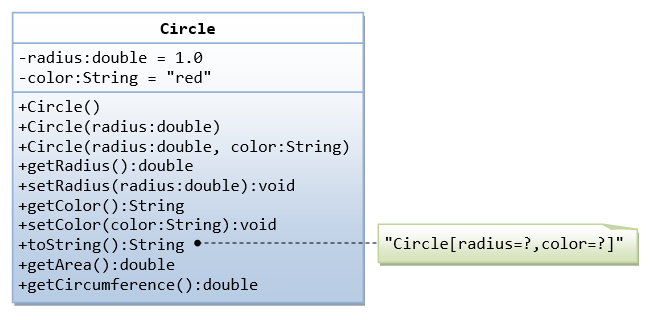
let p1 = new person("josh","23","MCA","abc@gmail.com","Ambattur-Chennai",9876543212);

OUTPUT:

object

|  |  |
| --- | --- |
| name | josh |
| age | 23 |
| degree | MCA |
| email | abc@gmail.com |
| address | Ambattur-Chennai |
| mobilenumber | 9876543212 |

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

[](https://github.com/rvsp/typescript-oops/blob/master/images/ClassDiagram_Circle.png)

**//getRadius():**

const pi=3.1412;

class circle {

constructor(radius,color)

{

this.radius=radius;

this.color=color;

}

getradius(){

return this.radius\*this.radius;

}

}

let c1= new circle(1.0,"red");

console.log(c1.getradius());

OUTPUT:1

**//getcolor():**

const pi=3.1412;

class circle {

constructor(radius,color)

{

this.radius=radius;

this.color=color;

}

getcolor(){

return this.color;

}

}

let c1= new circle(1.0,"red");

console.log(c1.getcolor());

OUTPUT: red

**//getArea():**

const pi=3.1412;

class circle {

constructor(radius,color)

{

this.radius=radius;

this.color=color;

}

getarea(){

return pi\*this.radius\*this.radius;

}

}

let c1= new circle(1.0,"red");

console.log(c1.getarea());

OUTPUT:

3.1412

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”

const PGrating="PG13";

class movie {

constructor(title,studio,rating)

{

this.title=title;

this.studio=studio;

this.rating=rating;

}

getPGrating(){

return this.rating+this.title;

}

}

let m1= new movie("Casino Royale","Eon Production","PG13");

console.log(m1.getPGrating());

OUTPUT: Casino Royale PG13

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

|  |  |
| --- | --- |
| Base fare | ₹60 |
| Distance fare (upto 20kms) | ₹12/km |
| Distance fare (after 20kms) | ₹18/km |
| Per minute fare | ₹2 |

//get dis1

var dis1="12/km";

Var dis2=”18/km”;

class uberprice{

constructor(basefare,perminfare)

{

this.basefare=basefare;

this.perminfare=perminfare;

}

getprice(){

return dis1\*this.basefare\*this.perminfare/100;

}

}

let u1= new uberprice(“60”,”2”);

console.log(m1.getdis1());