|  |
| --- |
| 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) |
|  |  |
|  | 1)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. |
|  |  |
|  | code- |
|  | class movie{ |
|  | constructor(title,studio,rating){ |
|  | this.title=title; |
|  | this.studio=studio; |
|  | this.rating=rating; |
|  | } |
|  |  |
|  | } |
|  | var movie1=new movie("maharshi","ramstudio","pg13"); |
|  | var movie2=new movie('bahubali','studio','r'); |
|  | console.log(movie1,movie2); |
|  |  |
|  | Output: |
|  |  |
|  | movie { title: 'maharshi', studio: 'wstudio', rating: 'pg13' } movie { title: 'bahubali', studio: 'studio', rating: 'r' } |
|  |  |
|  | 1)B)The constructor for the class Movie will set the class property rating to "PG" as default when no rating provided. |
|  |  |
|  | code- |
|  | class movie{ |
|  | constructor(title,studio,rating='PG'){ |
|  | this.title=title; |
|  | this.studio=studio; |
|  | this.rating=rating; |
|  | } |
|  |  |
|  | } |
|  | var movie1=new movie("kgf","studio"); |
|  | var movie2=new movie('svp2','A1studio','r'); |
|  | console.log(movie1,movie2); |
|  |  |
|  |  |
|  | Output: |
|  | movie { title: 'kgf', studio: 'studio', rating: 'PG' } movie { title: 'svp2', studio: 'A', rating: 'r' } |
|  |  |
|  |  |
|  |  |
|  |  |
|  | 1)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”. |
|  |  |
|  |  |
|  | code- |
|  | 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 { |
|  | title: 'Casino Royale', |
|  | studio: 'Eon Productions', |
|  | rating: 'PG13' |
|  | } |
|  |  |
|  | 2)converting uml class diagram t= class. |
|  |  |
|  | code- |
|  | 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(" radius of the circle is "+this.radius) |
|  | } |
|  | setradius(newradius){ |
|  | this.radius=newradius; |
|  |  |
|  | } |
|  | getcolor(){ |
|  | console.log(' color of the circle is '+this.color) |
|  | } |
|  |  |
|  | setcolor(newcolor){ |
|  | this.color=newcolor; |
|  | } |
|  | getarea(){ |
|  | console.log('area of circle is '+(Math.PI\*Math.pow(this.radius,2))) |
|  | } |
|  | getcircumference(){ |
|  | console.log(' circumference of a circle is '+(2\*Math.PI\*this.radius)) |
|  | } |
|  |  |
|  | } |
|  | let circle1=new circle(1,"green"); |
|  |  |
|  | circle1.getradius(); |
|  | circle1.setradius(3) |
|  | circle1.getradius(); |
|  | circle1.getcolor(); |
|  | circle1.setcolor("black"); |
|  | circle1.getcolor(); |
|  | circle1.getarea(); |
|  | circle1.getcircumference(); |
|  |  |
|  | Output: |
|  | radius of the circle is 1 |
|  | radius of the circle is 3 |
|  | color of the circle is green |
|  | color of the circle is black |
|  | area of circle is 28.274333882308138 |
|  | circumference of a circle is 18.84955592153876 |
|  | Execution Time: |
|  | 0.066s |
|  | Memory Used: |
|  | 8084kb |
|  |  |
|  | 3)Write a “person” class to hold all the details. |
|  |  |
|  | code- |
|  | class person { |
|  | constructor(name,father\_name,aadhar\_no,mother\_name,gender,address,mobile\_No,Email) { |
|  | this.name = name ; |
|  | this.father\_name = father\_name ; |
|  | this.mother\_name = mother\_name; |
|  | this.aadhar\_no = aadhar\_no; |
|  | this.gender = gender; ; |
|  | this.address = address ; |
|  | this.mobile\_No = mobile\_No; ; |
|  | this.Email = Email; |
|  | } |
|  | person\_name() { |
|  | return `my name is ${this.name}`; |
|  | } |
|  |  |
|  | person\_father\_name() { |
|  | return `my father name is ${this.father\_name}`; |
|  | } |
|  | person\_mother\_name() { |
|  | return `my mother name is ${this.mother\_name}`; |
|  | } |
|  | person\_aadhar\_no() { |
|  | return `aadhar\_no is ${this.aadhar\_no}`; |
|  | } |
|  | person\_gender() { |
|  | return `i am ${this.gender}`; |
|  | } |
|  | person\_address() { |
|  | return `my addressis ${this.adress}`; |
|  | } |
|  | person\_mobile\_no() { |
|  | return `my mobile\_no is ${this.mobile\_No}`; |
|  | } |
|  | person\_Email() { |
|  | return `my email is ${this.Email}`; |
|  | } |
|  | } |
|  | //object |
|  | const sai\_obj = new person("sai","srinivasulu","padma","2413-4132-9809","M","hyd","123456","sai@gmail.com"); |
|  |  |
|  | console.log(sai\_obj.person\_name()); |
|  | console.log(sai\_obj.person\_father\_name()); |
|  | console.log(sai\_obj.person\_mother\_name()); |
|  | console.log(sai\_obj.person\_aadhar\_no()); |
|  |  |
|  | console.log(sai\_obj.person\_gender()); |
|  | console.log(sai\_obj.person\_address()); |
|  | console.log(sai\_obj.person\_mobile\_no()); |
|  | console.log(sai\_obj.person\_Email()); |
|  |  |
|  | output- |
|  | my name is sai |
|  | my father name is srinivsulu |
|  | my mother name is padma |
|  | my aadhar no is 2413-4132-9809 |
|  | i am male |
|  | my adress is hyd |
|  | my mobile no is 123456 |
|  | my email is sai@gmail.com |
|  |  |
|  |  |
|  | 4)write a class to calculate uber price. |
|  |  |
|  | code- |
|  | class UberPriceCalculator{ |
|  | constructor(CRD,SBM,basefare,CPM,TR,ridedistance,bookingfee){ |
|  |  |
|  | this.CRD=CRD; |
|  | this.SBM=SBM; |
|  | this.basefare=basefare; |
|  | this.CPM=CPM; |
|  | this.TR=TR; |
|  | this.ridedistance=ridedistance |
|  | this.bookingfee=bookingfee; |
|  |  |
|  | } |
|  | getprice(){ |
|  | console.log(" UBER PRICE IS "+(this.basefare + ((this.CPM \* this.TR) + (this.CRD\* this.ridedistance) \*this.SBM) + this.bookingfee )) |
|  | } |
|  |  |
|  | setprice(newCRD,newSBM,newbasefare,newCPM,newTR,newridedistance,newbookingfee){ |
|  |  |
|  | this.CRD=newCRD; |
|  | this.SBM=newSBM; |
|  | this.basefare=newbasefare; |
|  | this.CPM=newCPM; |
|  | this.TR=newTR; |
|  | this.ridedistance=newridedistance |
|  | this.bookingfee=newbookingfee; |
|  |  |
|  | } |
|  | } |
|  | var cost1=new UberPriceCalculator(); |
|  | cost1.setprice(100,5,20,15,10,3,20) |
|  | cost1.getprice() |
|  |  |
|  | Output: |
|  | UBER PRICE IS 1690 |
|  | Execution Time: |
|  | 0.067s |
|  | Memory Used: |
|  | 8096kb |