

class Rectangle{

constructor(width,height){

this.width = width;

this.height = height;

}

}

let myRect1 = new Rectangle(10,5);

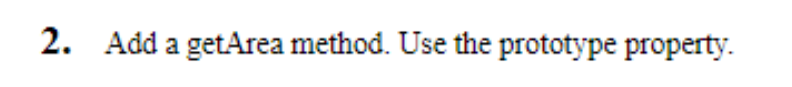
let myRect2 = new Rectangle(3,9);

let myRect3 = new Rectangle(39,45);

console.log(`Width of the Rectangle is : ${myRect1.width} and Height of the Rectangle is : ${myRect1.height}`)

console.log(`Width of the Rectangle is : ${myRect2.width} and Height of the Rectangle is : ${myRect2.height}`)

console.log(`Width of the Rectangle is : ${myRect3.width} and Height of the Rectangle is : ${myRect3.height}`)



class Rectangle{

constructor(width,height){

this.width = width;

this.height = height;

} }

Rectangle.prototype.getArea = function() {

return this.width \* this.height;

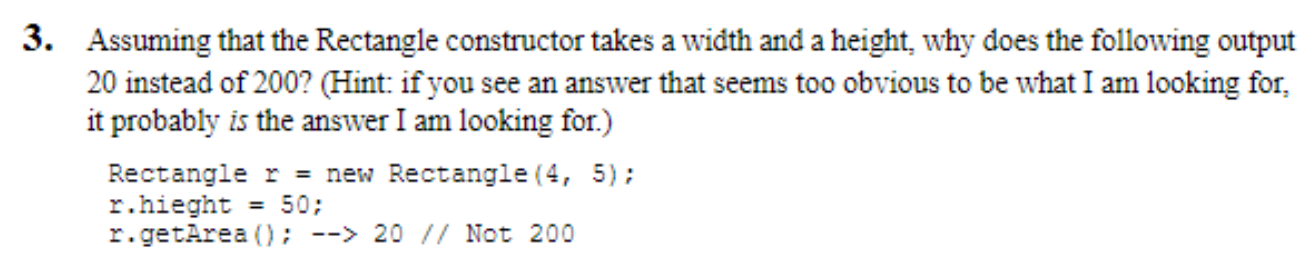
};

let myRect1 = new Rectangle(10,5);

let myRect2 = new Rectangle(3,9);

console.log(`Area Of Rectangle is : ${myRect1.getArea()}`)

console.log(`Area Of Rectangle is : ${myRect2.getArea()}`)



class Rectangle{

constructor(width,height){

this.width = width;

this.height = height;

} }

Rectangle.prototype.getArea = function() {

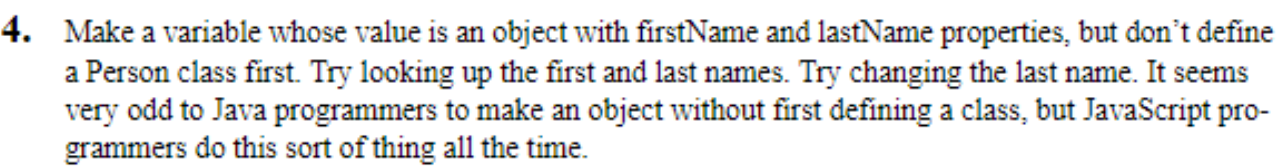
return this.width \* this.height;

};

let myRect1 = new Rectangle(4,5);

myRect.height = 50;

console.log(`Area Of Rectangle is : ${myRect1.getArea()}`)



var person ={

firstname : "Kunal",

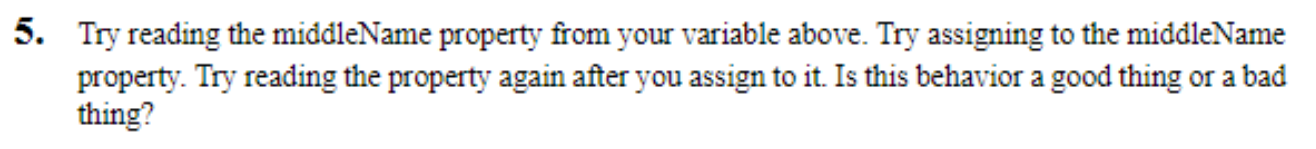
lastname : "Gupta"

}

console.log(person.firstname,person.lastname)

person.lastname = "Sharma";

console.log(person.firstname,person.lastname)



var person ={

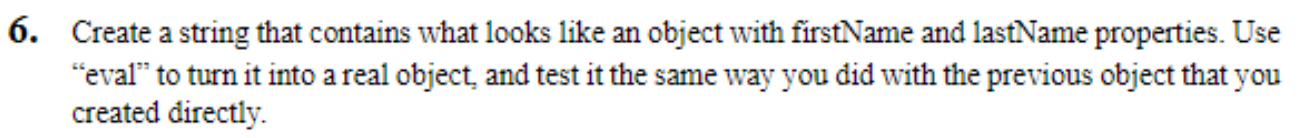
firstname : "Kunal",

lastname : "Gupta"

}

person.middlename = "Chand";

console.log(person.middlename);

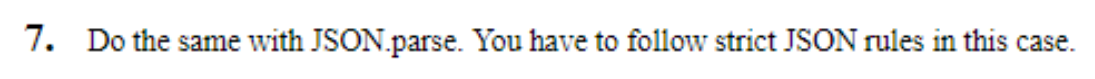


var mystr = '{ Firstname : "Kunal" , Lastname : "Gupta" }';

var str\_to\_obj = eval("(" + mystr + ")");

console.log(str\_to\_obj.Firstname);

console.log(str\_to\_obj.Lastname);



var person =JSON.parse('{"fname" : "Kunal","lname" :"Gupta","age":21,"skills":["C","C++","Python"],"address":{"city":"Alwar","pincode":301001},"dob":"24/10/1996","married":false,"profession":"analyst"}')

console.log(person.fname);

4 .

class BankAccount{

constructor(accountNumber,accountHolderName,accountBalance){

this.accountNumber = accountNumber;

this.accountHolderName = accountHolderName;

this.accountBalance = accountBalance;

}

}

class savingAccount extends BankAccount{

constructor(accountNumber,accountHolderName,accountBalance,isSalary)

{

super(accountNumber,accountHolderName,accountBalance);

this.isSalary = isSalary;

}

withdraw(amount){

if (amount<=this.accountBalance)

{

this.accountBalance = this.accountBalance-amount;

console.log("Thanks you");

// console.log("Your Balance is : " + this.accountBalance)

}else{

console.log("Insufficient Balance")

}

}

getCurrentBalance(){

console.log("Your Svaing Account Balance is : " + this.accountBalance)

}

}

class currentAccount extends BankAccount{

constructor(accountNumber,accountHolderName,accountBalance,odlimit)

{

super(accountNumber,accountHolderName,accountBalance);

this.odlimit = odlimit;

}

withdraw(amount){

if (amount<=this.accountBalance)

{

if(this.accountBalance-amount>this.odlimit){

this.accountBalance = this.accountBalance-amount;

console.log("Thanks you");

//console.log("Your Balance is : " + this.accountBalance)

}else{

console.log("limit extended")

}

}else{

console.log("Insufficient Balance")

}

}

getCurrentBalance(){

console.log("Your current Account Balance is : " + this.accountBalance) }

}

let mysavingAccount = new savingAccount(1110001102,"Kunal",300000,true);

let mycurrentAccount = new currentAccount(1110001102,"Kunal",300000,50000);

mysavingAccount.withdraw(24000)

mycurrentAccount.withdraw(38000)

mysavingAccount.getCurrentBalance()

mycurrentAccount.getCurrentBalance()