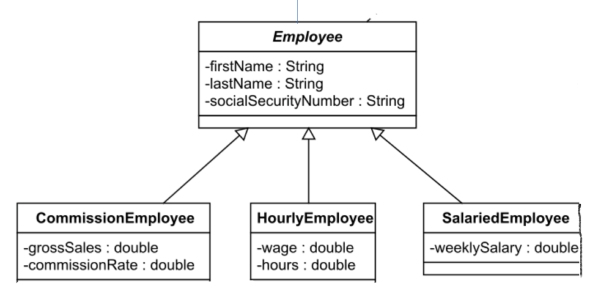
**I.TOPIC OBJECT ORIENTED PROGRAMMING**

**Q1.Given the following requirement write a python program using OOPs**



\*Include salary as 4rth attribute in Employee class

\*Note: contructor method init is common in all classes and is used to initialize the datamembers

1.Employee class methods:

Calculate salary: displays the message “ This method will be overridden”

Display : displays first name, lastname and social security number

2. CommissionEmployee methods:

Calculate salary: calculates salary as product of grossSales and commissionRate

Display : calls the display method in parent class using super keyword and also

displays salary for CommissionEmployee

3.HourlyEmployee methods:

Calculate salary: calculates salary as product of wage and hours

Display : calls the display method in parent class using super keyword and also

displays salary for HourlyEmployee

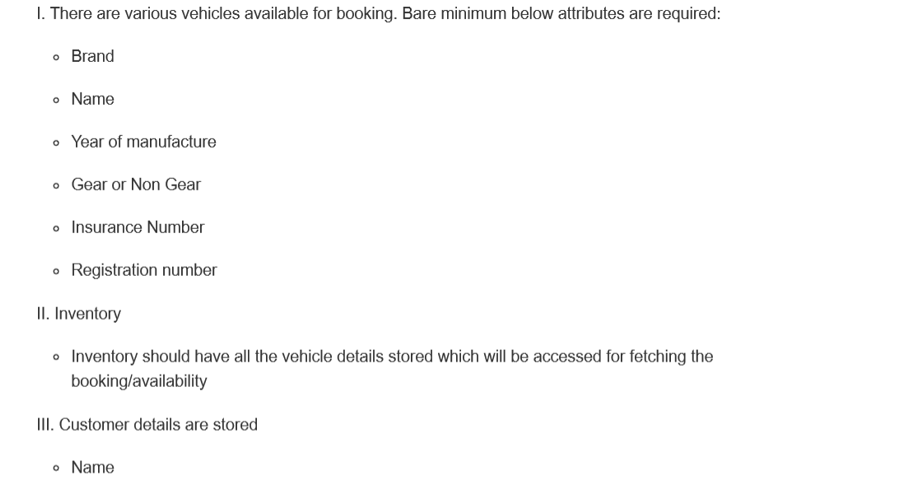
4.SalariedEmployee methods:

Calculate salary: calculates salary same as weekly salary

Display : calls the display method in parent class using super keyword and also

displays salary for SalariedEmployee

**Q2.Create a Two wheeler rental system with the following scenarios**



* Email
* Phone number
* Age
* Driving Licence
* KYC Done or not

**II.REGULAR EXPRESSIONS**

**Q1**. Read the special characters from a string, store the characters/occurrence as a key/value pair in a

dictionary. Sort the dictionary by value and invert the dictionary where the occurrence is the key and

the characters is the value. Finally count the occurance of each special character and show the

output in the following pattern.

**Sample output:**

Sample input

string='%%$@$^\_#)^)&!\_+]!\*@&^}@@%%+$&[(\_@%+%$\*^@$^!+]!&#)\*}{}}!}]$[%}@[{@#\_^{\*'

output={ 1:['('], 3:['#',')',']','[','{'], 4:['\_','&','+','\*'], 5:['!'],6:['$','^','}'], 7:['%'], 8:['@'] }

Q2.Write a logic using RegEx to parse any URL into components such as

