1. Make a class called *Member*. Create two attributes - *first\_name* and *last\_name*, and then create 5 more attributes that are typically stored in a Member profile, such as *email, gender, contact\_number,* etc. Make a method called *show\_info()* that prints a summary of the member’s information. Make another method called *get\_memeber()* that prints personalized greetings to the member.

Create 5 instances representing different members and call both the methods for each member.

1. Make a class called *Car*. The \_\_*init\_\_()* method for *Car* should store the following four attributes: *car\_model*, *no\_of\_gears*, *is\_started*, *is\_stopped*. Make a method *describe\_car()* that prints these four bits of information, and with the following methods:
2. *start\_car()*: if *Car* is already started, it should print the message “Car already started”, otherwise the message should be “Car started”.
3. *stop\_car()*: if *Car* is already stopped, it should print the message “Car already stopped”, otherwise the message should be “Cart stopped”.
4. *change\_gear()*: If *Car* has reached the maximum gear, it should print the message “Car is already in top gear”, otherwise it should increment the gear by one and the following message should be displayed “Car is moving at {#gear} gear”.

Create 3 instances representing *Car* and test each of the 3 methods.

1. Make a class called *Question*. The \_\_*init\_\_()* method for *Question* should store the following attributes: *questiondata*, *option1*, *option2*, *option3*, *option4*, *answer* and *questionlist*. Make the following methods:
2. *initialize()*:

Initialize 6 questions and store in *questionlist*. These can be random questions like “Who won the cricket world cup in 1983?”, “What is the name of the current PM of India?”, etc. Each of these questions should be stored in *questiondata, option1* thru *option4* can be 4 possible answers, and the correct answer should be stored in *answer.*

1. *beginTest(s)*:

//It should display questions from *questionlist* one by one, and has to prompt for the correct answer from the user. User input can be “A”, ”B”, ”C”, or ”D” and it should validate the input answer against the correct answer. If the input answer is correct, then the variable *ccount* has to be incremented by 1, else the variable *wcount* should be incremented by 1.

1. *showResult()*:

It should display the result of the test as shown in the example below.

Example:

Total questions: (value)

# of Correct answers: (value)

# of Wrong answers: (value)

Result: Pass/Fail (>40% is Pass, else Fail)

Take the test and test the application.

1. Make a class called *Employee.* It should be an abstract class with the following attributes and methods:

Attributes: *empno*, *ename*, *salary*

Methods: *cal\_salary()* [abstract method]

*display()* should display information *empno*, *ename*

Make a subclass of *Employee* with the name *PermanentEmployee* with the following attributes and methods:

Attributes: *empno*, *ename*, *basic\_salry*, *bouns*

Methods: *cal\_salary()* : [basic\_salary + bouns]

*display()* should display information *empno*, *ename*, *salary*

Make another subclass of *Employee* with the name *ContractEmployee* with the following attributes and methods:

Attributes: *empno*, *ename*, *day\_sal*, *days\_worked*

Methods: *cal\_salary()*: [day\_sal \* days\_worked]

*display()* should display information *empno*, *ename*, *salary*

Add at least 5 employees of different types and test the application.