AI-ML Course - Week 2 - Part 2- Assignment - OOP

Instructions:

- ✓ Please don't use ChatGPT or google to find answer. You can research on google, but don't copy paste code sample.
- ✓ First – brainstorm, what is problem definition. What is required? Think deep.
- ✓ Then prepare, logic , think- how to solve it. What data type – will be used? What is data structure – that will be used. Prepare logic – in words.
- ✓ Once you have full grip on problem definition, solution steps, then code
- ✓ For each task, copy question first in comments, then write logic
- ✓ For each task category, like "String Operations", create a separate file.
- ✓ Most task- are very simple. Task, just need 15 to 20 minutes for each task to do it. Practice and practice, to grip good, the topics under discussion.
- ✓ Side by side check-in code in your GitHub code repository. Don't share URL your GitHub in group, but share in 1to1 whatsapp chat with me, don't share in group, to avoid other student influenced by your code.
- ✓ Let's put best efforts.

Thanks

Object oriented programming – OOP Assignment

Question 1:

Write a Python program to create a person class. Include attributes like name, country and date of birth. Implement a method to determine the person's age.

Questions 2:

Write a Python program to create a calculator class. Include methods for basic arithmetic operations. Create a class calculator with methods of adding, subtracting, multiply, and divide.

Questions 3:

First create a abstract class, ICaculator, all abstract methods of operations, then do inheritance, to implement a calculator functions in derived class.

Questions 4:

Write a Python program to create a Data class for customer with following fields.

"Record ID","First Name","Last Name","Email","Phone Number","Contact owner","Primary Associated Company ID","Last Activity Date","Lead Status","Marketing contact status","Create Date","Associated Company"

Also, create 5 objects of this data class.

Question 5.

Write a Python program to create a class representing a stack data structure. Include methods for pushing and popping elements.

Example:

```
class Stack:
    def __init__(self):
        self.items = []
```

Question 6.

Write a Python program to create a class representing a queue data structure. Include methods for enqueueing and dequeueing elements.

class Queue:

```
def __init__(self):
 self.items = []
```

Questions 7.

Employee Class with Salary, Department, and Overtime Calculation

Write a Python class Employee with attributes like emp_id, emp_name, emp_salary, and emp_department and methods like calculate_emp_salary, emp_assign_department, and print_employee_details.

Sample Employee Data:

"ADAMS", "E7876", 50000, "ACCOUNTING"

"JONES", "E7499", 45000, "RESEARCH"

"MARTIN", "E7900", 50000, "SALES"

"SMITH", "E7698", 55000, "OPERATIONS"

Use 'assign_department' method to change the department of an employee.

Use 'print_employee_details' method to print the details of an employee.

Use 'calculate_emp_salary' method takes two arguments: salary and hours_worked, which is the number of hours worked by the employee. If the number of hours worked is more than 50, the method computes overtime and adds it to the salary. Overtime is calculated as following formula:

overtime = hours_worked - 50

Overtime amount = (overtime * (salary / 50))

Be creative in defining the class.

Questions 8.

Restaurant Class with Menu, Table Reservation, and Order Management

Write a Python class Restaurant with attributes like menu_items, book_table, and customer_orders, and methods like add_item_to_menu, book_tables, and customer_order.

Perform the following tasks now:

Now add items to the menu.

Make table reservations.

Take customer orders.

Print the menu.

Print table reservations.

Print customer orders.

Note: Use dictionaries and lists to store the data.

Be creative in defining the class.

Question 9.

BankAccount Class with Deposit, Withdrawal, and Transaction History

Write a Python class BankAccount with attributes like account_number, balance, date_of_opening and customer_name, and methods like deposit, withdraw, and check_balance.

Question 10. Create two separate programs, to use multiple inherence.

==========

Universe >> Earth >> Asia >> Pakistan
>> Korea
>> India

>> NorthAmerica >> USA
>> Canada

Jupiter

==========

Employee >> Manager >>CXO >> CEO >> Founder-Chairman
>> Finance Manager >> CIO
>> Operation Manager >> CTO
>> IT Manager

Ensure that all 5 types of inheritance, is implemented, in above 2 examples. Be creative in defining the class.

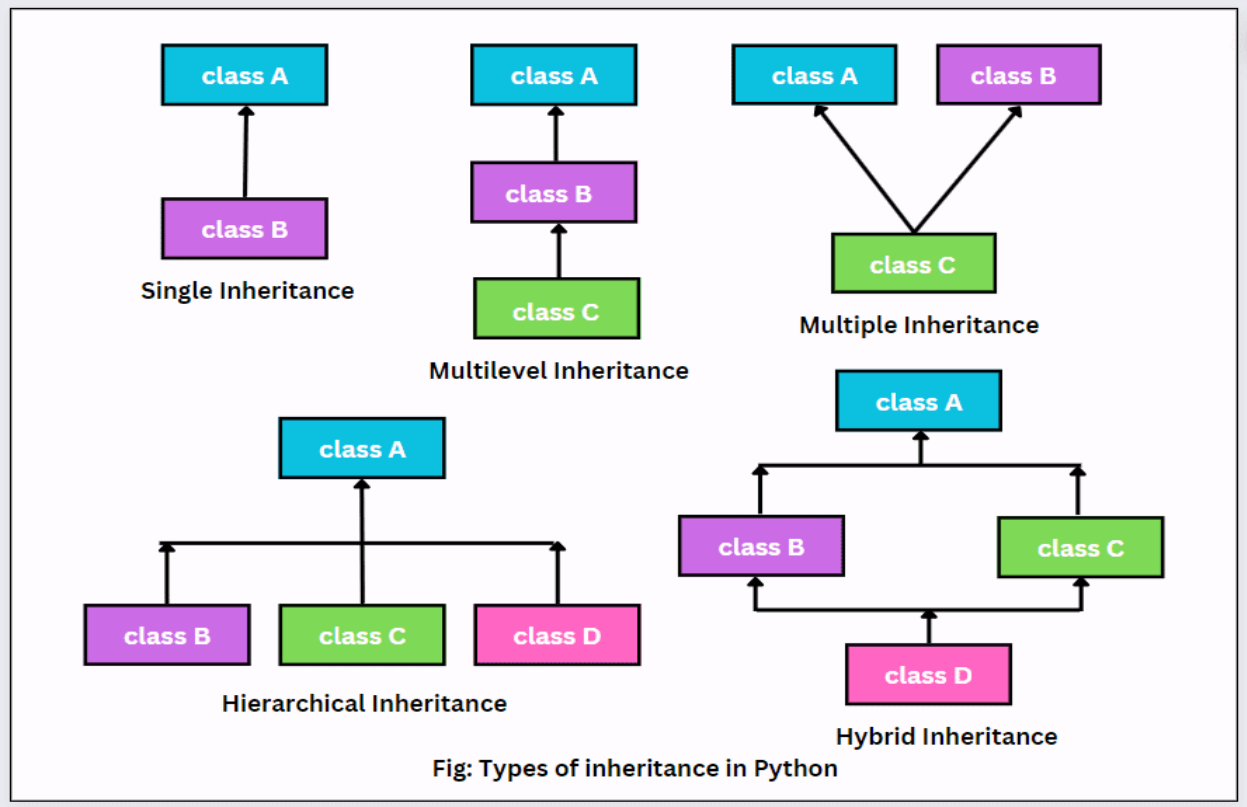


Fig: Types of inheritance in Python

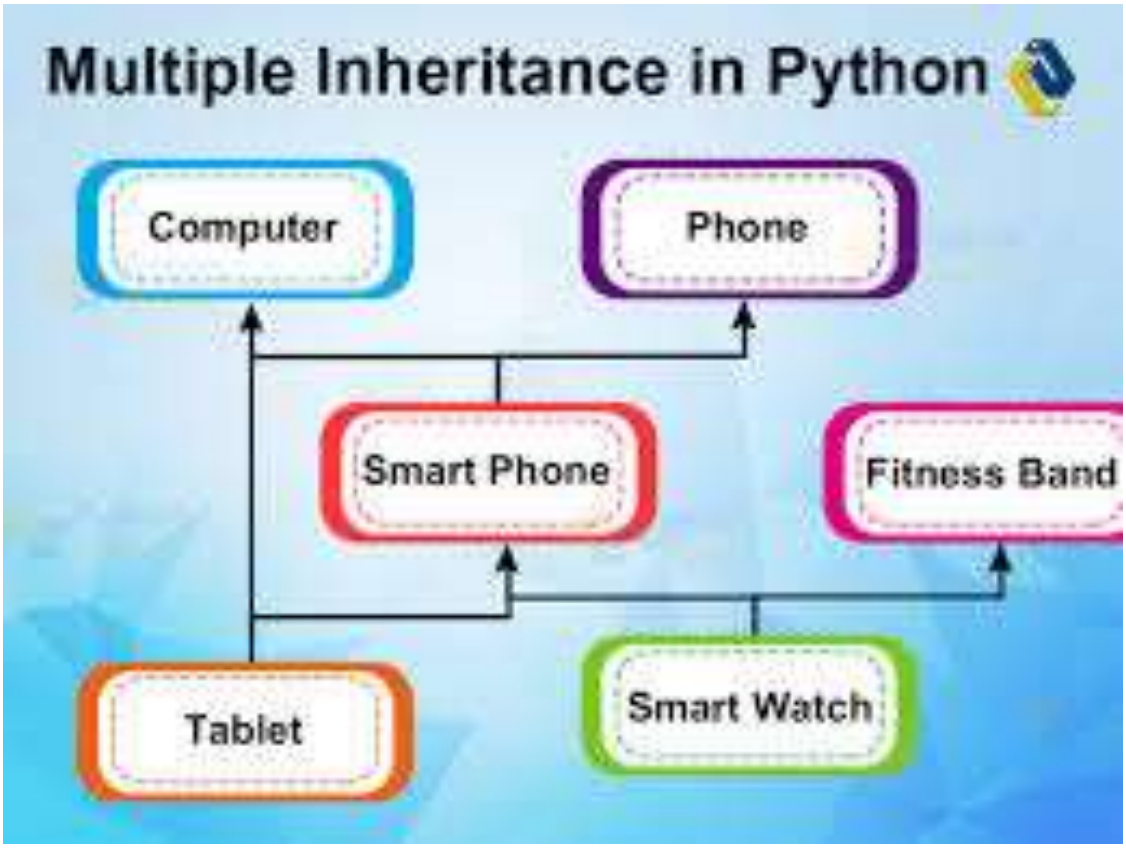
Questions 11:

Define following class hierarchy. Be creative in defining the classes.

Create - instance available and class available in all classes.

Create class, instance method , Static method and class method.

Create class objects, play with properties. Be creative in defining the classes.



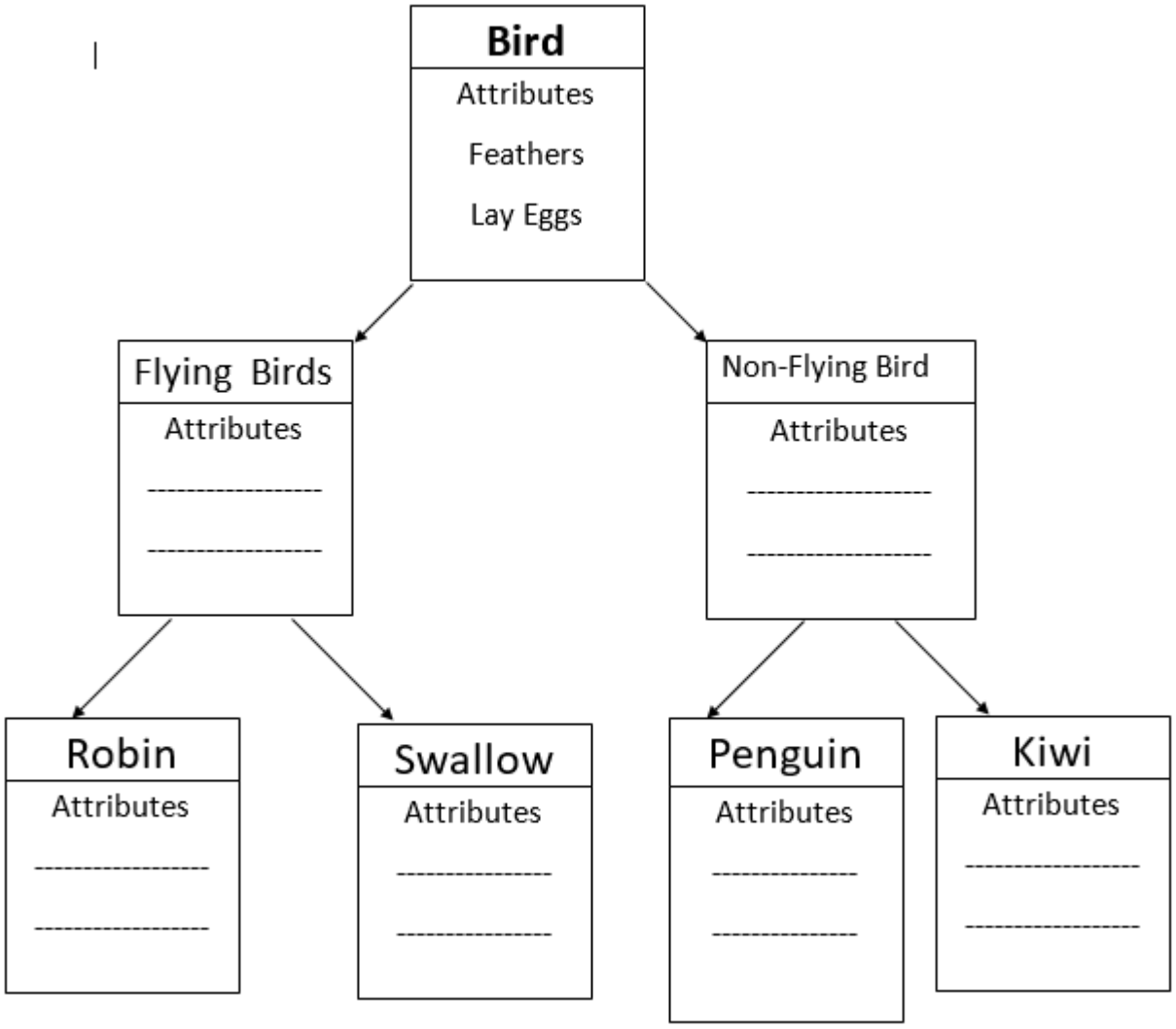
Question 12:

Define following class hierarchy. Be creative in defining the classes.

Create - instance available and class available in all classes.

Create class, instance method , Static method and class method.

Create class objects, play with properties. Be creative in defining the classes.



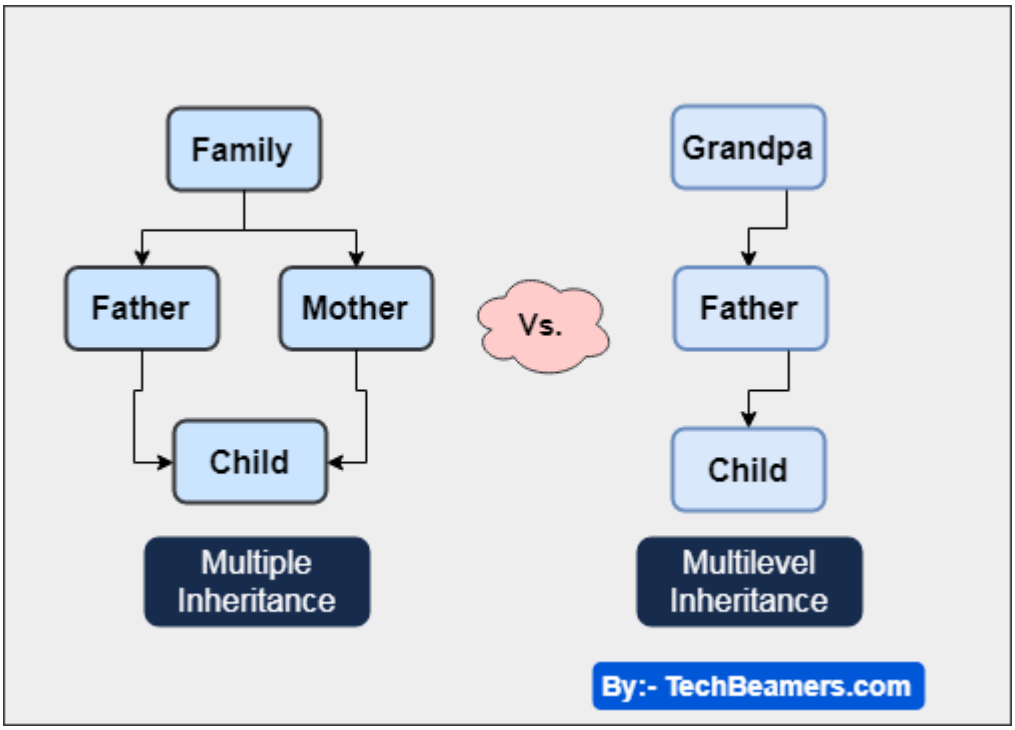
Question 13:

Define following class hierarchy. Be creative in defining the classes.

Create - instance available and class available in all classes.

Create class, instance method , Static method and class method.

Create class objects, play with properties. Be creative in defining the classes.



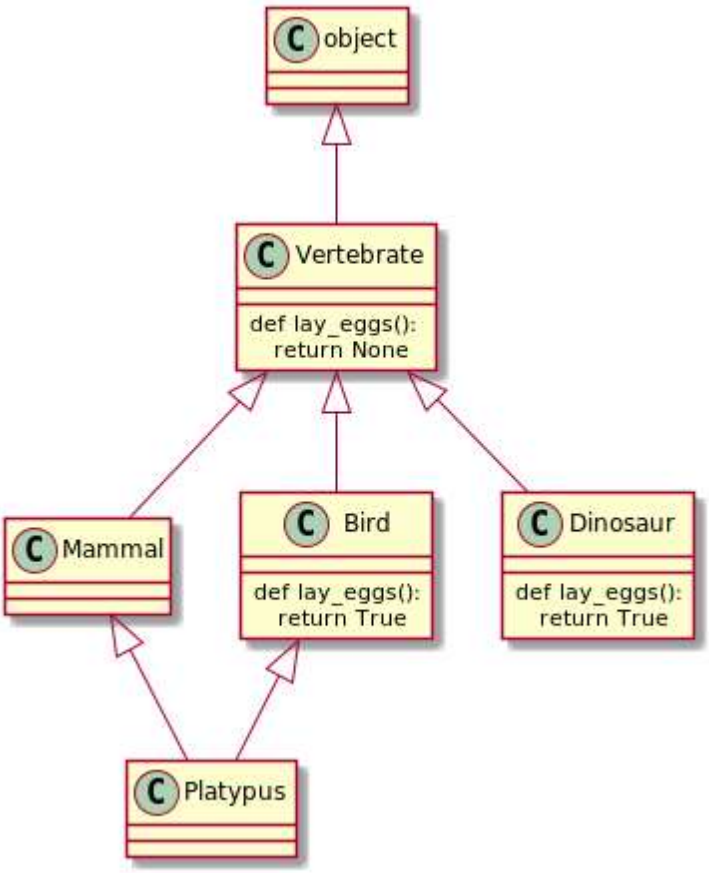
Question 14.

Define following class hierarchy. Be creative in defining the classes.

Create - instance available and class available in all classes.

Create class, instance method , Static method and class method.

Create class objects, play with properties. Be creative in defining the classes.



Question 15.

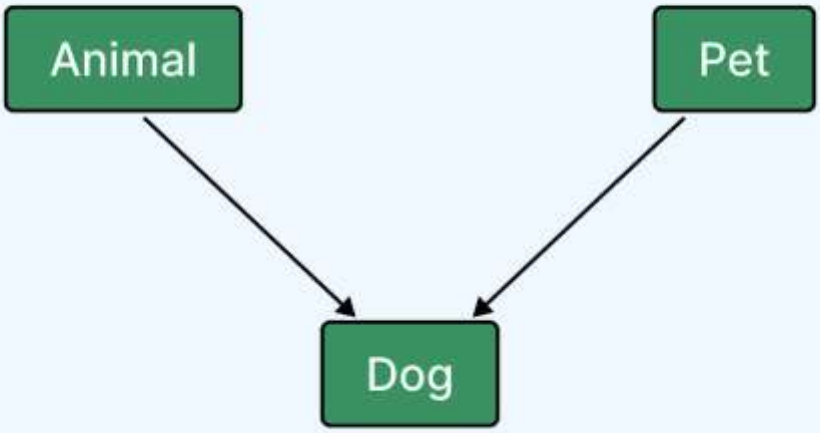
Define following class hierarchy. Be creative in defining the classes.

Create - instance available and class available in all classes.

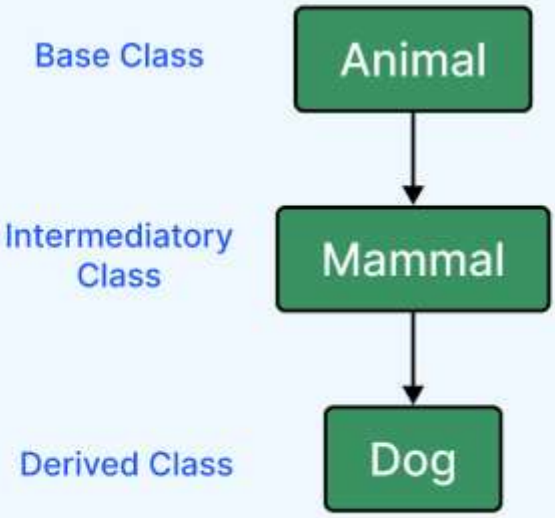
Create class, instance method , Static method and class method.

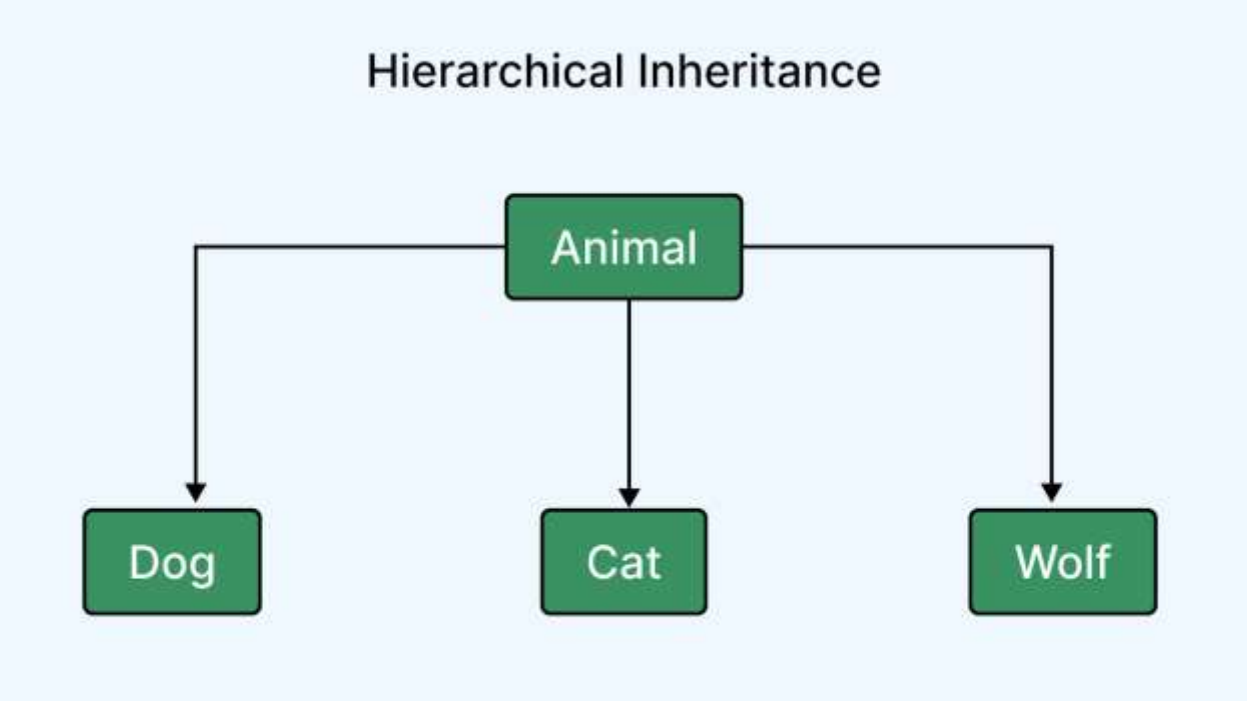
Create class objects, play with properties. Be creative in defining the classes.

Multiple Inheritance



Multilevel Inheritance





Note: Ensure that all 5 types of inheritance, is implemented, in above all Inheritance questions.. Be creative in defining the class.