OOPS

1. What is OOPS
2. How many types of principles in OOPS and what are they.
3. What is OOPL
4. What is the difference between OOPS and OOPL
5. Is Python supports OOPS
6. Is OOPS Principles fallows any order and what is that.
7. What is Abstraction
8. What is Encapsulation
9. How to achieve encapsulation in Python?
10. What is Inheritance
11. What is Polymorphism
12. What is class?
13. How to define a class?
14. What is Object?
15. What is the use of the Object?
16. How to create Object?
17. How to destroy or delete an Object from memory?

18. Where Object gets memory?
19. Where class gets memory?
20. What is instance variable?
21. What is static variable?
22. Why Instance variables?
23. Why static variables?
24. What is the difference between instance and static variables?
25. When static variables get memory?
26. When instance variables get memory?
27. Where static variables get memory?
28. Where instance variables get memory?
29. What are the ways of accessing static variables?
30. What are the ways of accessing instance variables?
31. What happens if I use all variables as instance in the class?
32. What happens if I use all variables as static in the class?
33. What is a method?
34. How to define a method in python?
35. How to use/call a method?
36. What is the use of method parameters?

37. What are the types of methods in python
38. When should I define an Instance method?
39. When should I define a static method?
40. What are the ways of accessing an instance method?
41. What are the ways of accessing a static method?
42. What are the variables can instance method access
43. What are the variables can static method access
44. Can we create a variable inside the method?
45. Can we print a variable without setting a value? If not why?
46. Who will provide the initial/default value to local variable?
47. Can we initialize the instance or static variables through method?
48. Can we assign the instance or static variables through method?
49. Who will provide the initial/default value to instance variable?
50. Can a method call another method?
51. Can I call a static method from instance method?
52. Can I call an instance method from instance method?
53. Can I call an instance method from static method?

54. Can I call a static method from static method?
55. What is constructor?
56. What are the rules in Defining a constructor?
57. Where/When Constructor is used?
58. Can I write a class without defining at least constructor?
59. Can I create an object if a class is having zero constructors?
60. What are the types of constructor?
61. What is the parameterized constructor?
62. When should we use parameterized constructor?
63. How to call a constructor?
64. Can we use a constructor to initialize static variables?
65. Can we call a constructor from a method(static/instance)?
66. Can we call a method from constructor?
67. Can constructors access static variables?
68. Can constructors access instance variables?
69. Can I declare constructor as static
70. Can I write a method with class name?
71. What happens if I define a method with class name?
72. What is Inheritance?
73. What is the use of Inheritance?

74. Can static members participate in inheritance?

75. Can instance members participate in inheritance?

76. Can we inherit a constructor?

77. What are the types of Inheritance?

78. What happens if we create an Object to sub class/any class?

79. How to initialize the super class instance variables?

80. How to call a super class constructor?

81. Who will call upper class constructor by default?

82. How to call one constructor from another constructor of same class?

83. What is the use of calling one constructor from another of same class?

84. What is Overloading?

85. What is method overloading?

86. What is constructor overloading?

87. What is the use of overloading?

88. Can we overload a constructor in different class?

89. Can we overload an instance method in different class?

90. Can we overload a constructor in same class?

91. Can we overload an instance method in same class?

92. Can we overload variables?

93. Can we overload static method?

94. Can we overload instance method?

95. What is overriding?

96. What is method overriding?

97. What is constructor overriding?

98. Is overriding done without inheritance?

99. What is the use of overriding?

100. Dose overriding depends on parameter?

101. Dose overriding depends on method name?

102. Can we overriding methods in same class?

103. Can we override methods in different class?

104. Can we override variables?

105. How to stop method overriding?

OOPS Case Studies

1. Write a class Define a class to represent a bank account. Include the following members:

Data Variables : Name of the Depositor, Account Number, Type of Account, Balance amount in the account.

Data Methods : To assign initial values, To deposit an amount, To withdraw an amount after checking the balance, To display name and balance.

2. Create a class called Student with the following details:

Roll No, Stud Name, Marks In Eng, Marks In Maths, Marks In Science. Display the total marks and Percentage of the student.

3. Change the above class definition so that you can calculate marks for five students.

4. For an Online Bookstore create a class to store book details and display the book details with fields are book number, book name, book title, book author, quantity of books, book price. calculate and display the bill amount .

5. Create a reference type called Person. Populate the Person class with the following attributes to store the following information: First name, Last name, Email address, Date of birth Add constructors that accept the following parameter lists: All four parameters First name , Last name , Email ,First name , Last name , Date of birth

write appropriate methods to accept and display the details.

6. Case Study:

Yatra.Com Travels is a committed tour and travel company. It has devised many innovative packages for its customers who want to take a holiday. There are three kinds of tours:

a)Discover India b)Holiday Hungama c)Pilgrimage Package

These tours start every week. A customer can avail of a package belonging to any category, starting on any given date. The customer can also specify the number of people accompanying the customer. The customer class will have the following member data: customer name, number of people accompanying(int), package category(D/H/P), cost(float), tour start date

Write an executable program with a class called Customer with required attributes and methods.

7. Consider the following scenario:

A Furniture Manufacturer manufactures domestic furniture. Customers provide their specifications to the company for the furniture they want. To cope up with the received customer's order, FFC decides to computerize the order-processing system. The System should accept the values of furniture items, such as a bookshelf and a chair. You need to develop the hierarchy of these items.

8. Create the classes required to store data regarding different types of Courses. All courses have name, duration and course fee. Some courses are part time where you have to store the timing for course. Some courses are onsite where you have to store the company name and the no. of candidates for the course. For onsite course we charge 10% more on the course fee. For part-time course, we offer 10% discount.

Provide constructors and the following methods.

Print() GetTotalFee()

9. Create a class to store details of student like rollno, name, course joined and fee paid so far. Assume courses are Core Pyhon and Advance Python with course fees being 3000 and 3500.

Provide a constructor to take rollno, name and course. Provide the following methods: Payment(amount) ,Print() ,DueAmount property, Total Fee property

Add a static member to store Service Tax, which is set to 12.3%. Also allow a property through which we can set and get service tax.

Modify Total Fee and Due Amount properties to consider service tax

10. Create a class called Customer with two methods

customertype() which displays the type of customer getPriviledge() which displays the privileges according to type of the customer.

Create a class called CorporateCustomer which inherits the Customer class and overides the methods given in the Customer class.

Create a class called PersonalCustomer which inherits the Customer class and overides the methods given in the Customer class.

create another class called MainClass to execute the program.

11. Write an executable program.

Create a class Shape with the following methods

getDetails() to get details from the user

calculateArea() to calculate the area with the given dimensions

displayDetails() to display the calculated area of the shape.

Create a class called Triangle which inherits Shape class and provides appropriate implementation of the methods given in the base class.

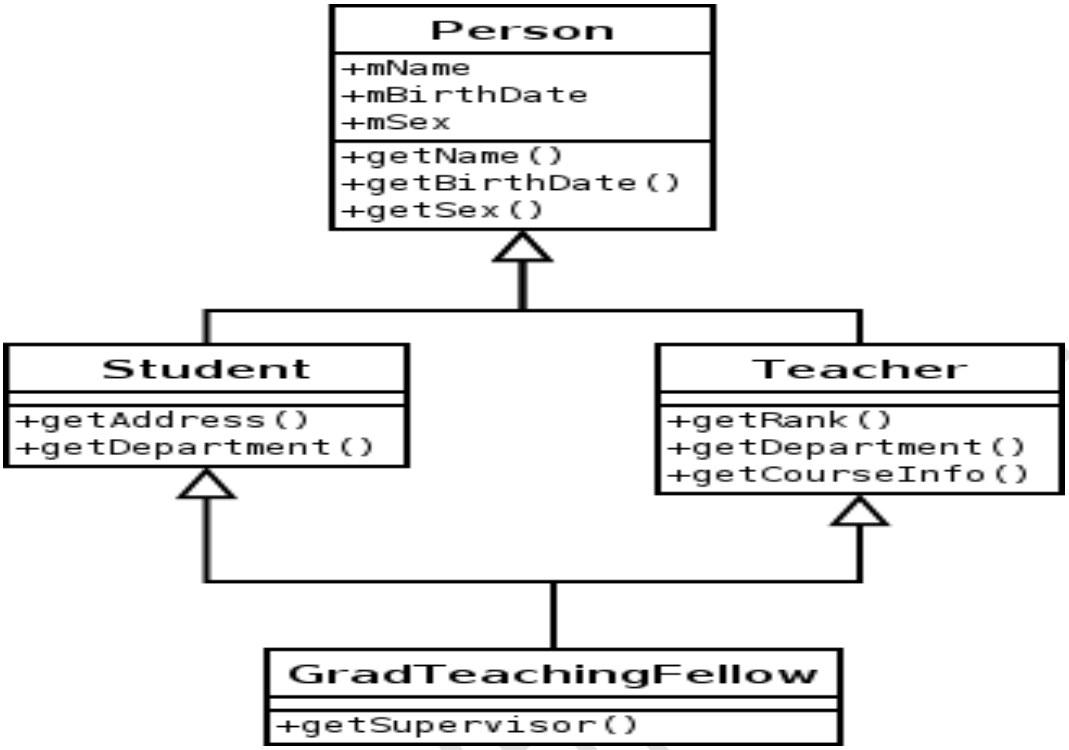
Create a class called Circle which inherits Shape class and provides appropriate implementation of the

12. Martin wants to create a ticket booking application for a movie theater. The application should ask the user for his choice, whether he wants to book the tickets or not. The application should also ask the user for the total number of tickets to be booked. While booking the tickets if the total number of booked tickets exceeds the available tickets, the application should raise an exception.

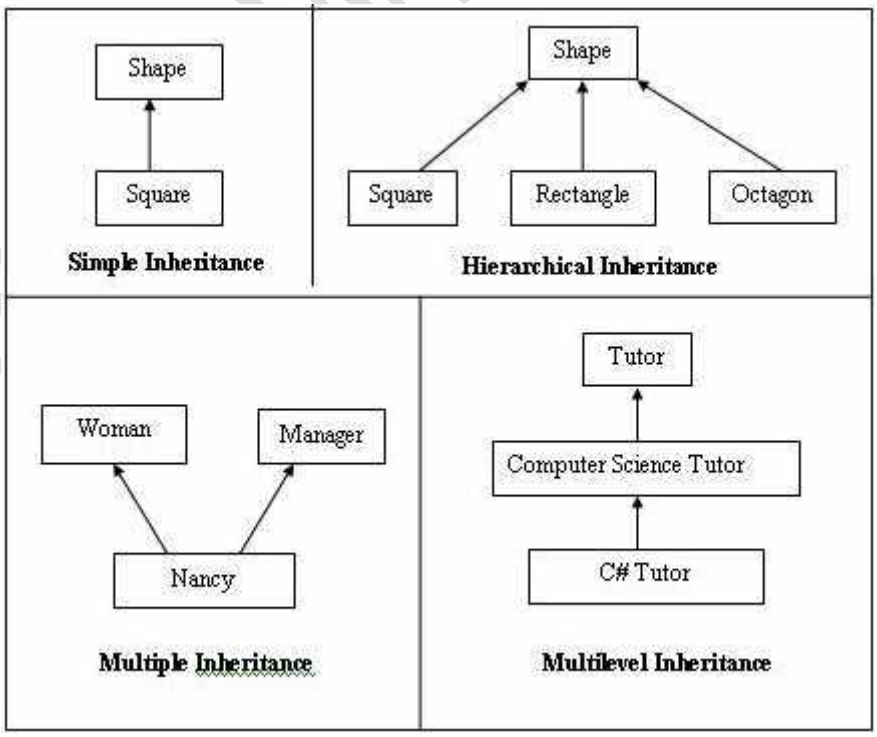
Assume the total number of available tickets is 15.

13. Create a base class, Telephone, and derive a class Electronic Phone from it. In Telephone, create a protected string member phonetype, and a public method Ring( ) that outputs a text message like this: "Ringing the <phonetype>." In ElectronicPhone, the constructor should set the phonetype to "Digital." In the Run( ) method, call Ring( ) on the Electronic Phone to test the **inheritance**?

14. Implement the Following Multiple Inheritance Program.



15. WAP to implement inheritance?



16. Create a class to implement shapes and draw the following shapes

a) Rectangle

b) Circle

c) Triangle

And Find their Areas Using Method Area.

17. Create a class to implement calculate simple and compound interest using method find interest method.

18. Create a class to Calculate Employee Total Salary Accept Basic Salary, DA, HRA from keyboard If Employee Salary Greater Then 20,000 Then increment 20% of salary and Display Total Salary

Overloading & overriding:

19.Wap on Overloading to Implement Addition operations with different Parameters ex : add(int,int), add(float,int), add(float,int,str);

20. Wap on Overriding Methods To implement Display method In parent class and child class with same signature.