SET-1

1. Create a **Bus** child class that inherits from the Vehicle class. The default fare charge of any vehicle is **seating capacity * 100**. If Vehicle is **Bus** instance, we need to add an extra 10% on full fare as a maintenance charge. So total fare for bus instance will become the **final amount = total fare + 10% of the total fare.**

Note: The bus seating capacity is **50**. so the final fare amount should be **5500.** You need to override the fare() method of a Vehicle class in Bus class.
Use the following code for your parent Vehicle class. We need to access the parent class from inside a method of a child class.

```
class Vehicle:
 def __init__(self, name, mileage, capacity):
 self.name = name
 self.mileage = mileage
 self.capacity = capacity

 def fare(self):
 return self.capacity * 100

class Bus(Vehicle):
 pass

School_bus = Bus("School Volvo", 12, 50)
print("Total Bus fare is:", School_bus.fare())
```

Expected Output:

Total Bus fare is: 5500.0

2) From this below mentioned code snipppets

```python
class Book(object):
    def __init__(self, authorlast, authorfirst, \
                 title, place, publisher, year):
        self.authorlast = authorlast
        self.authorfirst = authorfirst
        self.title = title
        self.place = place
        self.publisher = publisher
        self.year = year

    def write_bib_entry(self):
        return self.authorlast \
            + ', ' + self.authorfirst \
            + ', ' + self.title \
            + ', ' + self.place \
            + ':  ' + self.publisher + ', ' \
            + self.year + '.'

beauty = Book( "Dubay", "Thomas" \
             , "The Evidential Power of Beauty" \
             , "San Francisco" \
             , "Ignatius Press", "1999" )
pynut = Book( "Martelli", "Alex" \
            , "Python in a Nutshell" \
            , "Sebastopol, CA" \
            , "O'Reilly Media, Inc.", "2003" )
```

Consider the Book definition given in Example
Here are some questions to test your understanding of what it does: 1. How would you print out the author attribute of the pynut instance (at the interpreter, after running the file)?

2. If you type print beauty.write bib entry() at the interpreter (after running the file), what will happen?

   3. How would you change the publication year for the beauty book to "2010"?

3) Define a class named CaseString with the following methods:
• the constructor takes an initial string as a parameter
• set_upper: sets string mode (False = original case, True = upper case) •
set_string: sets the encapsulated string
• get_string: returns the string with the proper case

4) Define a class named Circle which can be constructed by a radius. The Circle class has two methods for computing perimeter and area, respectively.

5) Define a class named Shape and its subclass Square. The Square class has a constructor which takes a length as argument. Both classes have an area function which can print the area of the shape where Shape's area is 0 by default.

SET-2

1.Write a Python class to find the two elements that sum to zero from a set of n real numbers. Go to the editor
Input array : [-25, -10, -7, 7, 10, -3, 8, 3]
Output : [[-10,10], [-7,7], [-3,3]]

2.Write a Python class which has two funtions get_Str and print_Str. get_Str accept a string from the user and print_Str print the string in upper case and lower case.

3.Write a Python class named rectangle constructed by a length, breadth, and two methods which will compute the area and the perimeter of a rectangle.

4.A vehicle class consist of vehicle data and standard methods to manipulate vehicle data. Write a Python child class Bus that will inherit all of the variables and methods of the Vehicle class. Child class consist of new methods that update the value of vehicle class and display in the output.

5. Friend of two are jointly planning for the startups, during month one their discussion was about the, "1. Infrastructure of the work area", "2. Capital and funding" and "3. Human resources". Three months later they started implementing. Represent this concept by using Abstract class.

SET-3

1. Design a program of class Car with some attributes and its object to print its attributes.
2. Develop a program of class Room with attributes length, breadth and height and its object room1 and room2 to calculate the area and volume of a room using function.
3. A farmer wants to build a wooden fence around a rectangular field. He measures the length and the width of the field and decides how high the fence should be. He alsos decides how wide the space between each board of the fence should be.Each board is 10cm wide. Help him with a object oriented program that calculates the total length of all boards required to be bought.
4. Make a class Employee with a name and salary. Make a class Manager inherit from Employee. Add an instance variable, named department, of type string. Supply a method to String that prints the manager's name, department and salary. Make a class Executive inherits from Manager. Supply a method to String that prints the string "Executive" followed by the information stored in the Manager superclass object.

Supply a test program that tests these classes and methods.

5. Imagine a tollbooth with a class called toll Booth. The two data items are a type unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both these to 0. A member function called payingCar ( ) increments the car total and adds 0.50 to the cash total. Another function, called nopayCar ( ), increments the car total but adds nothing to the cash total. Finally, a member function called displays the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car.

SET-4

APP Week3: OOPs

1. Find whether the student is eligible for the current year placement from the inputs student name, CGPA, gender (M/F), number of backlogs. Implement the above by using object oriented programming concept.

2. Using constructor, find the total EB bill to be paid. Rule slab is given below.

Units 0-100 - Rs.0

Units 101-200 - Rs.1.00

Units 201-300 - Rs. 2.00

Units 301 - more - Rs. 3.00

Input:

Previous reading: 5897

Current reading: 6121

Output:

Bill amount: 148

3. Two friends are playing book cricket, the possible page numbers during the random flips could be 0, 2,4,6,8, as a rule, if the player gets page number 0 then he is out and if the player gets page number 8 then he has got 1 run. Find who the winner is. Implement it by using polymorphism.

Input:

friend1: 8 8 2 4 8 8 0

friend2: 6 6

Output:

Winner: friend2

4. Every year, to calculate income tax for the employee, it has been very hectic for the accounts department, so they are planning to issue a new card called "IT card" which will merge the details of aadhar card, pan card and faculty id card. Implement this logic by using inheritance.

5. Friend of two are jointly planning for the startups, during month one their discussion was about the, "1. Infrastructure of the work area", "2. Capital and funding" and "3. Human resources". Three months later they started implementing. Represent this concept by using Abstract class.

SET-5

1. Employee payroll system
   Employee name.
   Employee ID
   Basic pay
   HRA
   TA
   DA
   Grade pay
   Deduction 5% from Basic pay

   Find the Gross pay, Net pay (Grade pay + deductions) and print all the details given above, using class and object of get () method and cal( )method.

0. Student management system
   Student name
   Student id as inputs with marks for 5 Subjects, calculate and round of the subject marks to 10 and then find CGPA, Grade using object and class concept of oops
   Solve the above by creating a base class and derived class use inheritance to solve it
   .
0. Instantiate the X class using self and then using self in the get () method and to achieve composition you can instantiate other objects in the class and then use those instances that Generate output of the total code which has both inheritance and composition .
0. An Internet service provider has three different subscription packages for its customers: Package A: for $9.95/month, 10 hours of access are provided. Additional hours are $2.00/hour. Package B: for $14.95/month, 20 hours of access are provided. Additional hours are $1.00/hour. Package C: for $19.95/month, unlimited access is provided.
   Write a program using objects and classes that calculates a customer's monthly bill. It should ask which package the customer has purchased and how many hours were used. It
   should then display the total amount due.

Input validation: be sure the user only selects package A, B, or C. Also, the number of hours used in a month cannot exceed 744.

0. Write a program using object-oriented concepts that calculates and displays a person's body mass index (BMI). The

BMI is often used to determine whether a person with sedentary lifestyle is overweight or underweight for his or her height. A person's BMI is calculated with the following formula:

$BMI = weight * 703/height^2$

where weight is measured in pounds and height is measured in inches. The program should display a message indicating whether the person has optimal weight, is underweight, or is overweight. A sedentary person's weight is considered to be optimal if his or her BMI is between 18.5 and 25. If the BMI is less than 18.5, the person is considered to be underweight. If the BMI value is greater than 25, the person is considered to be overweight.

SET-6

1. Develop an Python code to display the following output using class and object (only one class and two objects)

Blu is a bird
Woo is also a bird
Blu is 10 years old
Woo is 15 years old

2. Develop an python code for the following scenario, Class D is an derived class which has the properties class B via class B1 and class B2, create an object to call the member functions of B, B1 and B2 also sketch the what type of inheritance it belongs to.

3. Develop an python code to restrict access to methods and variables (Program using private data member, assign the value 900 to private data member and try to change the value, show whether your new value is changed or not).
4. Develop an python code to call the function fly which exist in both the class parrot and penguin, how will you achieve using common interface flying test.

5. Develop an python code to explain how single inheritance will perform using super() function.

SET-7

1. Write a python program to count number of objects in a class

2. Create a Vehicle class with max_speed and mileage instance attributes. 3. Create child class Bus that will inherit all of the variables and methods of the Vehicle class.

4. Create a Btech_SecondYear_CS class where the constructor function sorts the student name based on their percentages of attendance (using quick sort). After displaying the student list, the destructor function will be called for free the memory.

5. Write a python program to implement queue structure using class.

SET-8

1. Create a Vehicle class with max_speed and mileage instance attributes 2. Create child class Bus that will inherit all of the variables and methods of the Vehicle class

3. Create a Bus class that inherits from the Vehicle class. Give the capacity argument of Bus.seating_capacity() a default value of 50.

4. Create a Bus child class that inherits from the Vehicle class. The default fare charge of any vehicle is seating capacity * 100. If Vehicle is Bus instance, we need to add an extra 10% on full fare as a maintenance charge. So total fare for bus instance will become the final amount = total fare + 10% of the total fare.

5. Determine which class a given Bus object belongs to (Check type of an object)

```
class Vehicle:
 def __init__(self, name, mileage, capacity):
 self.name = name
 self.mileage = mileage
 self.capacity = capacity

 class Bus(Vehicle):
 pass

 School_bus = Bus("School Volvo", 12, 50)
```

SET-9

1. Write a Python program to show that the variables with a value assigned in class declaration, are class variables and variables inside methods and constructors are instance variables.

2. Write a Python program to show that we can create instance variables inside methods

3. Write a python code to implement following scenario, In a company Factory, staff and Office staff have certain common properties - all have a name, designation, age etc. Thus they can be grouped under a class called CompanyMember. Apart from sharing those common features, each subclass has its own characteristic - FactoryStaff gets overtime allowance while OfficeStaff gets traveling allowance for an office job. The derived classes (FactoryStaff & OfficeStaff) has its own characteristic and, in addition, they inherit the properties of the base class (CompanyMember). 4. Write a python program to access hidden variable outside the class using object and which threw an exception.

5. Write a Python program to demonstrate that hidden members can be accessed outside a class

SET-10

1. Mr.Kannan had two shops. His son Sid inherited all the shops; he decided to change Shop1 according to his aspirations. Shop2 is a family business inherited from forefathers, so Mr.Kannan strictly prohibited changes to it. Shop3 is a new shop opened by Sid himself. Illustrate this situation using inheritance.

0. Write a Python class to find the three elements that sum to zero from a set of n real numbers
Input array : [-25, -10, -7, -3, 2, 4, 8, 10]
Output : [[-10, 2, 8], [-7, -3, 10]]

0. Mr. Zourist — a very popular programmer — wrote a program in Zotlin with N classes numbered from 1 to N. Some of these classes have an implementation of a function F. The Zotlin compiler needs to determine which definition of function F should be called for an object of Class 1.

0. Write a program to create a room class, the attributes of this class is roomno, roomtype, roomarea and ACmachine. In this class the member functions are setdata and displaydata.

0. Write a program to create a class named shape. In this class we have three sub classes circle, triangle and square each class has two member function named draw () and erase (). Create these using polymorphism concepts.
SET-11

Problem 1:

Write a definition for a class named Kangaroo with the following methods: An

__init__ method that initializes an attribute named pouch_contents to an empty list.

A method named put_in_pouch that takes an object of any type and adds it to pouch_contents.

A __str__ method that returns a string representation of the Kangaroo object and the contents of the pouch.

Test your code by creating two Kangaroo objects, assigning them to variables named kanga and roo, and then adding roo to the contents of kanga's pouch.

Problem 2:

Create a class BookStore which uses to store the details of various books. included a constructor which initilaize the details of book such as title, author, publisher, and price of kind private instance variable. Include a class level variable noOfBooks which keep the count of books created. include the method BookInfo which describe the details of each Book. Create instance to create new books and display the details

Problem 3:

Create a Queue class. Queues are a fundamental computer science data structure. You can add elements to a queue, and they maintain a specific order. When you want to get something off the end of a queue, you get the item that has been in there the longest (this is known as first-in-first-out', or FIFO).

In your Queue class, you will need three methods:

• init : to initialize your Queue (think: how will you store the queue's elements? You'll need to initialize an appropriate object attribute in this method)

• insert: inserts one element in your Queue

• remove: removes one element from your Queue and returns it. If the queue is empty, return a message that says it is empty (without throwing an error that halts your code).

Note: Do not use Python's built in Queue data structure. Implement your own Queue with lists and class.

Problem 4:

Create a deck of cards class. Internally, the deck of cards should use another class, a card class. Your requirements are:
The Deck class should have a deal method to deal a single card from the deck. After a card is dealt, it is removed from the deck.

There should be a shuffle method which makes sure the deck of cards has all 52 cards and then rearranges them randomly.

The Card class should have a suit (Hearts, Diamonds, Clubs, Spades) and a value (A,2,3,4,5,6,7,8,9,10,J,Q,K)

Problem 5:

Create a class called Numbers, which has a single class attribute called MULTIPLIER, and a constructor which takes the parameters x and y (these should all be numbers).

Write a method called add which returns the sum of the attributes x and y.

Write a class method called multiply, which takes a single number parameter a and returns the product of a and MULTIPLIER.

Write a static method called subtract, which takes two number parameters, b and c, and returns b - c.

Write a method called value which returns a tuple containing the values of x and y. Make this method into a property, and write a setter and a deleter for manipulating the values of x and y.

SET-12

1. Create a class customer with name, address as its member variables. In list of account holder's information, find the account holder how live nearby (using pincodes). If a

name of the person is given then, customers of the same bank lives nearby should be given as output

0. Create a class that can calculate the IT the based on the loan and saving account transactions by the customer (Take the total amount deposited through out the year in an array, check the transaction which leads to saving in the same bank or other modes. Deduct the expenditure on saving to the maximum of 1.5L. For the rest of the amount calculate 6% of IT till 10L and 8% till 15L above that make 10% as IT).

0. Implement the following scenario using Inheritance. Create a class bank make appropriate variables, assign customer class as child of bank. Create classes Saving and Loan and Current account. Make all these classes as child of bank. Maintain transaction function in all these class to track the tractions

0. In continuation with the above bank class. Now create scoreboard for the customers. Function name of score same in all three classes but scores the point according to the account type.

Saving account : Score 1 pt for each 2k credit and deduct .25 for each 2k debit, 1 point for consistent balance of 10k

Current account: Score 1 pt for each 2k credit 1 point for consistent balance of 10k, reduce 1 pt for overdue more than 25k

Loan amount: Give 1 point for each on time repay and reduce 0.5 points for penalty payment.

0. Allow the customers to rate the services of the bank, give (display)customer a choice of service which they want to rate, and allow them to rate. Display the total score given by the particular customer and as the whole. (As an advancement take the score along with date display a chart with sum of scores on each period for each service).

SET-13

1. Write the python code to calculate the gross pay of an employee. Create a class name called employee and get the following inputs from the user.

- Name of the employee,
- Age,
- DOB,
- DOJ,
- Basic pay
- HRA
- Deductions

Print the gross salary of each employee.

0. Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.

    Sample Output: 120

0. Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.

    Sample Output: 200.96, 50.24

0. Write a Python class to find the three elements that sum to zero from a set of n real numbers.
    Input array : [-25, -10, -7, -3, 2, 4, 8, 10]
    Output : [[-10, 2, 8], [-7, -3, 10]]

0. Write a Python class to reverse a string word by word.

    Input string : 'hello .py'
    Expected Output : '.py hello'

SET-14

1. Write a Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number.

 Input: numbers= [10,20,10,40,50,60,70], target=50
Output: 3, 4

2. Write a Python class which has two methods get_String and print_String. get_String accept a string from the user and print_String print the string in upper case.

3. Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.

4. Write a Python class to find validity of a string of parentheses, '(', ')', '{', '}', '[' and ']'. These brackets must be close in the correct order, for example "()" and "()[]{}" are valid but "[)", "({[)]" and "{{{" are invalid.

5. Write a Python class to get all possible unique subsets from a set of distinct integers.
Input : [4, 5, 6]
Output : [[], [6], [5], [5, 6], [4], [4, 6], [4, 5], [4, 5, 6]]

SET-15

1. Aditya is the Strategist of the company. He just gives the skeleton of the process ideas and his team will have to implement it wherever applicable. One such idea is to give bonus of 10% for all along with hike based on the performance index. Create an interface and implement salary calculation method for Employees.

2. Write a Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number.
   Input: numbers= [10,20,10,40,50,60,70], target=50
   Output: 3, 4
3. Write a program suppose, it is required to build a project consisting of a number of classes, possibly using a large number of programmers. It is necessary to make sure that every class from which all other classes in the project will be inherited. Since any new classes in the project must inherit from the base class, programmers are not free to create a different interface. Therefore, it can be guaranteed that all the classes in the project will respond to the same debugging commands.
4. Write a Python class to reverse a string word by word. Input string : 'hello .py'
   Expected Output : '.py hello'
5. Define an interface using Java that contains method to calculate the perimeter of an object. Define two classes circle and rectangle with suitable fields and methods. Implement the interface "perimeter" in these classes. Write the appropriate main() method to create object of each class and test all the methods.

SET-16

1. Write a Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number.

 Input: numbers= [10,20,10,40,50,60,70], target=50
Output: 3, 4

2. Write a Python class to reverse a string word by word.

Input string : 'hello .py'
Expected Output : '.py hello'

3. Define a class named Shape and its subclass Square. The Square class has an init function which takes a length as argument. Both classes have a area function which can print the area of the shape where Shape's area is 0 by default.

4. Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area.
5.Write a Python class to find validity of a string of parentheses, '(', ')', '{', '}', '[' and ']'. These brackets must be close in the correct order, for example "()" and "()[]{}" are valid but "[)", "({[)]" and "{{{" are invalid.