Week3 Set15

Q1

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.

In [1]:

```
class Employee:
    def __init__(self, name, age, basic, grade):
        self.name = name
        self.age = age
        self.basic = basic
        self.hike = 1000
        self.grade = grade
    def salary(self):
        if self.grade > 6:
            self.salary = 1.1*self.basic + self.hike
        else:
            self.salary = self.basic
    def display(self):
        print('Employee name: ' + self.name)
        print('Employee age: ' + str(self.age))
        print('Employee Salary: ' + str(self.salary))

emp = Employee('Ram', 36, 20000, 7)
emp.salary()
emp.display()
```

```
Employee name: Ram
Employee age: 36
Employee Salary: 23000.0
```

Q2

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

In [3]:

```
class sumTarget:
    def twoSum(self, nums, target):
        arr = []
        for num in nums:
            if target-num in arr:
                return (num, target-num)
            else:
                arr.append(num)
print(sumTarget().twoSum([10,20,30,40,50,60,70], 50))
```

```
(30, 20)
```

Q3.

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.

In [8]:

```
# IMPLEMENTING INTERFACE
class Salary:
    def __init__(self, pay, bonus):
        self.pay = pay
        self.bonus = bonus
    def annualSalary(self):
        return (self.pay*12) + self.bonus


class Employee:
    def __init__(self, name, age, pay, bonus):
        self.name = name
        self.age= age
        self.objSalary = Salary(pay, bonus)
    def totalSalary(self):
        return self.objSalary.annualSalary()


emp = Employee('Max', 20, 15000, 2000)
print('Salary of the employee is: ' + str(emp.totalSalary()))
```

```
Salary of the employee is: 182000
```

Q4

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

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

In [6]:

```
class solution:
    def reverseWords(self, s):
        return ' '.join(reversed(s.split()))

string = input('Enter a valid string: ')
print(solution().reverseWords(string))
```

```
Enter a valid string: python .py
.py python
```

Q5

Define an interface using Python 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.

In [7]:

```
class Circle:
    def perimeter(self, r):
        print('The perimeter of the circle is: ' + str(2*3.14*r))

class Rectangle:
    def perimeter(self, l, b):
        print('The perimeter of the rectangle is: ' + str(2*(l+b)))

Circle().perimeter(25)
Rectangle().perimeter(10, 20)
```

```
The perimeter of the circle is: 157.0
The perimeter of the rectangle is: 60
```