1) Create a class named Shape. Add an instance method called area. But don't define the method, just raise NotImplemented() exception with a message.

In [55]:
```python
class shape:
    def area(self):
        raise NotImplementedError("Not Implemented The Method ")
shapes=shape()
shapes.area()
```

```
---------------------------------------------------------------------------
NotImplementedError                       Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_14584\768413728.py in <module>
      3         raise NotImplementedError("Not Implemented The Method ")
      4 shapes=shape()
----> 5 shapes.area()

~\AppData\Local\Temp\ipykernel_14584\768413728.py in area(self)
      1 class shape:
      2     def area(self):
----> 3         raise NotImplementedError("Not Implemented The Method ")
      4 shapes=shape()
      5 shapes.area()

NotImplementedError: Not Implemented The Method
```

2) Make it an abstract base class by inheriting ABC class from the abc module. (To import: from abc import ABC)

Make the area method an abstract method by decorating it with abstractmethod decorator (import this also from abc).

In [53]:
```python
from abc import ABC,abstractmethod
class shape(ABC):


    @abc.abstractmethod
    def area(self):
        pass
class shapes(shape):

    def area(self,area):
        print("Area is ",area)




shap=shapes()
shap.area(500) # 500 is just an example value given
```

```
Area is  500
```

3) Add 4 different subclasses for class Shape. - Triangle, Square, Pentagon, Circle.

Define custructor for each of them to assign the necessary parameters required for calculating the area. Define the area method for each of the classes. When invoked it Should return the area of the shape by calculating it.

In [65]:
```python
class shape:
    pass
class Triangle(shape):
    def __init__(self,a,b,c):
        self.a=a
        self.b=b
        self.c=c
    def area(self):
        s=self.a+self.b+self.c/2
        print("Area of the Triangle = ",s*(s-self.a)*(s-self.b)*(s-self.c)**.5)

class Square(shape):
    def __init__(self,hight):
        self.hight=hight

    def area(self):
        print("Area of the Square = ",self.hight*self.hight)

class Pentagon(shape):
    def __init__(self,a,b):
        self.a=a
        self.b=b
    def area(self):
        x=5*self.a*self.b
        print("Area Of The Pentagon = ",x/2)

class Circle(shape):
    def __init__(self,r):
        self.r=r

    def area(self):
        pi=3.14
        print("Area Of Circle = ",pi*self.r**2)



triangle=Triangle(10,20,30)
triangle.area()

square=Square(10)
square.area()

pentagon=Pentagon(10,20)
pentagon.area()

circle=Circle(10)
circle.area()
```

```
Area of the Triangle =  152498.71925691704
Area of the Square =  100
Area Of The Pentagon =  500.0
Area Of Circle =  314.0
```

4) Create a class Employee with name and id, salary attributes.

The salary has to be calculated and should be initialized to zero. Create a method to calculate the salary by taking the no of hours worked and multiply it by 200.You can give no of hours to the method as an argument.

Now create a child class ParttimeEmployee by inheriting the Employee class, and by using method overriding (create salary calculation method in this class also with the same name) get the salary of part time employee by multiplyig no of hours worked by 150. Also implement '+' operator overloading using **add** to find the total expense of paying salaries to employees(Find the total salary of all the created employees and parttime employees use **radd** and implement chained addition).

In [93]:
```python
class Employee:

    def __init__(self,name,ID,hour):
        self.name=name
        self.ID=ID
        self.hour=hour
        self.salary=hour*200
    def Calculate_Salary(self):

        print("Employee Name = ",self.name,"\nEmployee Id = ",self.ID,"\nWorking Hour = ",self.hour,
              "\nEmployee Salary = RS.",self.salary)
        print("_____________________________________________________________________________________")


class Part_Time_Employee(Employee):
    def __init__(self,name,ID,hour):
        self.name=name
        self.ID=ID
        self.hour=hour
        self.salary=hour*150
    def Calculate_Salary(self):
        salary=self.hour*150
        print("Part_Time_ Employee Name = ",self.name,"\nEmployee Id = ",self.ID,"\nWorking Hour = ",
              self.hour,"\nEmployee Salary = RS.",self.salary)
        print("_____________________________________________________________________________________")




emp1=Employee("ajith","007",15)
emp1.Calculate_Salary()

emp2=Employee("Anandu",123,10)
emp2.Calculate_Salary()

part_time_emp1=Part_Time_Employee("Akshay",178,20)
part_time_emp1.Calculate_Salary()

part_time_emp2=Part_Time_Employee("Soumya",143,15)
part_time_emp2.Calculate_Salary()


a=emp1.salary
b=emp2.salary
c=part_time_emp1.salary
d=part_time_emp2.salary


print("Employee 1 Salary = ",a)
print("Employee 2 Salary = ",b)
print("Employee 3 Salary = ",c)
print("Employee 4 Salary =",d)
Total=a.__add__(b+c+d)
print("Total Salary Expense = ",Total)
```

```
Employee Name =  ajith
Employee Id =  007
Working Hour =  15
Employee Salary = RS. 3000
_____________________________________________________________________________________
Employee Name =  Anandu
Employee Id =  123
Working Hour =  10
Employee Salary = RS. 2000
_____________________________________________________________________________________
Part_Time_ Employee Name =  Akshay
Employee Id =  178
Working Hour =  20
Employee Salary = RS. 3000
_____________________________________________________________________________________
Part_Time_ Employee Name =  Soumya
Employee Id =  143
Working Hour =  15
Employee Salary = RS. 2250
_____________________________________________________________________________________
Employee 1 Salary =  3000
Employee 2 Salary =  2000
Employee 3 Salary =  3000
Employee 4 Salary = 2250
Total Salary Expense =  10250
```

In [ ]: