

Task 36

```
from abc import ABC , abstractmethod
class Animal(ABC):
    @abstractmethod
    def sound(self):
        pass

class Lion(Animal):

    @property
    def sound(self):
        print("Sound from Lion")

class tiger(Animal):

    @property
    def sound(self):
        print("Sound from Tiger")

a1 = Lion()
a1.sound

a2 = tiger()
a2.sound
```

Task 37

```
import math

from abc import ABC , abstractmethod

class Shape(ABC):
    @abstractmethod
    def calculateArea(self):
        pass

class Circle(Shape):
    def __init__(self, Radius):
        self.Radius = int(Radius)

    @property
    def calculateArea(self):
        print("Area OF Circle :",math.pi * self.Radius**2)

class Shape(ABC):
    @abstractmethod
    def calculatePerimeter(self):
        pass

class Triangle(Shape):
    def __init__(self, a,b,c):
        self.a = a
        self.b = b
        self.c = c
    @property
    def calculatePerimeter(self):
        print("Triangle Of Perimeter :",self.a+self.b+self.c)

a1 = Circle(3)
a1.calculateArea
print("-----")
```

```
a2 = Triangle(3,4,5)
a2.calculatePerimeter
```