Execrise 1.
Build a program to manage a university's course catalog. You want to define a base class Course that has the following properties:
course_code: a string representing the course code (e.g., "CS101")
course_name: a string representing the course name (e.g., "Introduction to Computer Science")
credit_hours: an integer representing the credit hours for the course (e.g., 3)
You also want to define two subclasses CoreCourse and ElectiveCourse, which inherit from the Course class.
CoreCourse should have an additional property required_for_major which is a boolean representing whether the course is required for a particular major.
ElectiveCourse should have an additional property elective_type which is a string representing the type of elective (e.g., "general", "technical", "liberal arts")

```python
class course:
    def __init__(self,course_code,course_name,credit_hours):
        self.course_code = course_code
        self.course_name = course_name
        self.credit_hours = credit_hours

class CoreCourse(course):
    def __init__(self,course_code,course_name,credit_hours,required_for_major):
        super().__init__(course_code,course_name,credit_hours)
        self.required_for_major = required_for_major

class ElectiveCourse(course):
    def __init__(self,course_code,course_name,credit_hours,elective_type):
        super().__init__(course_code,course_name,credit_hours)
        self.elective_type = elective_type

x = CoreCourse ("CS101","Introduction in Computer Science",3,True)
y = ElectiveCourse ("ART101","Introduction to Arts",4,"Arts")
```

```python
print("CoreCourse ")
print(f'Code : {x.course_code}')
print(f'Code : {x.course_name}')
print(f'Credit Hours : {x.credit_hours}')
print(f'required_for_major : {x.required_for_major}')
print("\nElectiveCourse")
print(f'Code : {y.course_code}')
print(f'Code : {y.course_name}')
print(f'Credit Hours : {y.credit_hours}')
print(f'Elective Type : {y.elective_type}')
```

```
"C:\Users\VINOD VM\PycharmProjects\pythonProject\venv\Scr
CoreCourse
Code : CS101
Code : Introduction in Computer Science
Credit Hours : 3
required_for_major : True

ElectiveCourse
Code : ART101
Code : Introduction to Arts
Credit Hours : 4
Elective Type : Arts

Process finished with exit code 0
```

Exercise 2:
Create a base class Shape with methods area() and perimeter() and a subclass Rectangle that inherits from Shape and implements its own version of the area() and perimeter() methods.

```python
#Exercise 2
class Shapes:
    def area(self):
        pass
    def peremeter(self):
        pass

class Rectangle(Shapes) :
    def __init__(self,height,width):
        self.height = height
        self.width = width
    def area(self):
        return self.height * self.width

    def peremeter(self):
        return 2 * (self.height + self.width)
x = Rectangle (4,7)
print(f'Arear : {x.area()}')
print(f'Arear : {x.peremeter()}')
```

gle › peremeter()

Python_Class

```
"C:\Users\VINOD VM\PycharmProjects\pythonProject\venv\Script
Arear : 28
Arear : 22

Process finished with exit code 0
```

Exercise 3:
Create a base class Vehicle with properties make, model, and year and a method description() that returns a string with information about the vehicle. Create two subclasses Car and Motorcycle that inherit from Vehicle and implement their own version of the description() method.

```
class Vehicles:
    def __init__(self,make,model,year):
        self.make  =  make
        self.model  =  model
        self.year  =  year
    def description(self):
        return f'{self.make}{self.model}{self.year}'

class car(Vehicles):
    def description(self):
        return f'CAR : {self.make}{self.model}{self.year}'

class motorcycle(Vehicles)  :
    def description(self):
        return f'MOTORCYCLE : {self.make}{self.model}{self.year}'

c = car ("MARUTI ","VXI ",2013)
print(c.description())
m = motorcycle ("HERO HONDA ","PLUS ",2009)
print(m.description())
```

```
Python_Class
"C:\Users\VINOD VM\PycharmProjects\pythonProject\venv\Scripts\pytho
CAR : MARUTI VXI 2013
MOTORCYCLE : HERO HONDA PLUS 2009
```