الاسم : احمد السيد عبده على فرج | سكشن 1

Chapter 5

Rectangle Class

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

    def perimeter(self):
        return (self.width + self.height) * 2
# test
rec = Rectangle(5 , 2) # w = 5 , h = 2
print(rec.area()) #
print(rec.perimeter()) # 14
```

Employee Class

```
class Employee:
    def __init__(self , id , name , salary):
        self.employee_id = id
        self.name = name
        self.salary = salary

    @classmethod
    def from_string(cls , employee_str):
        vals = employee_str.split(',')
   return cls(vals[1] , vals[0] , vals[2])

    def display_employee_info(self):
        print(f"Id: {self.employee_id} , Name:
{self.name} , Salary: {self.salary}")


emp = Employee.from_string("John Doe,E123,50000")

emp.display_employee_info() # Id: E123 , Name: John Doe ,
Salary: 50000
```

Vehicle Hierarchy

```
class Vehicle:
    def move(self):
        print("vehicle is moving...")
class Car(Vehicle):
    def move(self):
        print("car is driving...")

class Bike(Vehicle):
    def move(self):
        print("bike is cycling...")

vehicles = [Vehicle() , Car() , Bike()]
for v in vehicles:
    v.move()
```

Vector

```
                                        class Vector:
                                def __init__(self , x , y):
                                            self.x = x
                                            self.y = y
                                 def __sub__(self , other):
                    return Vector(self.x - other.x , self.y -
                                              other.y)
                                 def __mul__(self , other):
                    return Vector(self.x * other.x , self.y *
                                              other.y)
                                      def printVec(self):
                           print(f"({self.x} , {self.y})")

                                       v1 = Vector(1 , 2)
                                       v2 = Vector(3 , 4)
                                            ex1 = v2 - v1
                                            ex2 = v2 * v1
                                    ex1.printVec() -> (2 ,2)
                                    ex2.printVec() -> (3 ,8)
```

Shape Polymorphism

```
import math

class Shape:
    def area(self):
```

```
        return 0

class Rectangle(Shape):
    def __init__(self , w , h):
        self.width = w
        self.height = h

    def area(self):
        return self.width * self.height


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

    def area(self):
        return math.pi * (self.radius ** 2)




# the main function:
def print_shape_area(shape):
    print(f"the area of this shape is
{shape.area()}")


shape = Shape()
rec = Rectangle(3 , 4)
circle = Circle(4)

print_shape_area(shape)
print_shape_area(rec)
print_shape_area(circle)
```