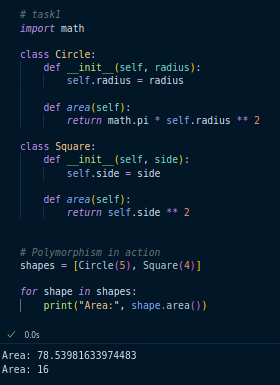
# Python OOP Practice Tasks

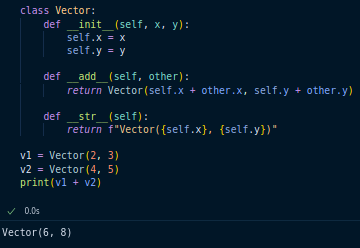
This document contains 10 tasks designed to practice Polymorphism, Operator Overloading, Magic/Dunder Functions, Dynamic Polymorphism, Abstract Classes, Empty Classes, Data Classes, and Keyword Arguments. The difficulty progresses from upper-basic to lower-advanced.

## Polymorphism & Operator Overloading

* Task 1: Create two classes Circle and Square with a method area(). Use polymorphism to call the area() method on both objects in a loop.



* Task 2: Create a Vector class with two attributes x and y. Implement operator overloading for + (using \_\_add\_\_) so that v1 + v2 adds their coordinates. Test with two Vector objects.



## Magic Functions / Dunder Functions

* Task 3: Create a class Book with attributes title and author. Implement \_\_str\_\_ so that printing the object shows: 'Book: Title by Author'. Implement \_\_len\_\_ to return the length of the title.

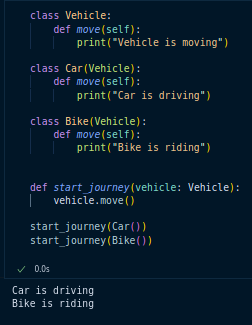


* Task 4: Create a class Employee with attributes name and salary. Overload the > operator (\_\_gt\_\_) to compare two employees by salary. Test by comparing two employees.

## 

## Dynamic Polymorphism (Subclass as Base Class)

* Task 5: Create a base class Vehicle with a method move(). Create subclasses Car and Bike, overriding move(). Write a function start\_journey(vehicle: Vehicle) that accepts any vehicle and calls move(). Test with both Car and Bike objects.



* Task 6: Implement a base class Shape with draw() method. Subclasses: Circle, Rectangle, Triangle. Write a loop that takes a list of mixed shapes and calls draw() on each.

## 

## Abstract Class / Empty Class / Data Class

* Task 7: Create an abstract class Appliance with an abstract method turn\_on(). Implement subclasses WashingMachine and Refrigerator. Each should implement turn\_on() differently.
* Task 8: Create an empty class Placeholder using pass. Dynamically add attributes name and value after creating the object. Print them.
* Task 9: Create a data class Student with fields name, age, and grade. Create 3 students and store them in a list. Loop through and print their details.

## Keyword Arguments

* Task 10: Write a function register\_employee(name, age, role, salary) that prints employee details. Call it once using positional arguments and once using keyword arguments in different order.