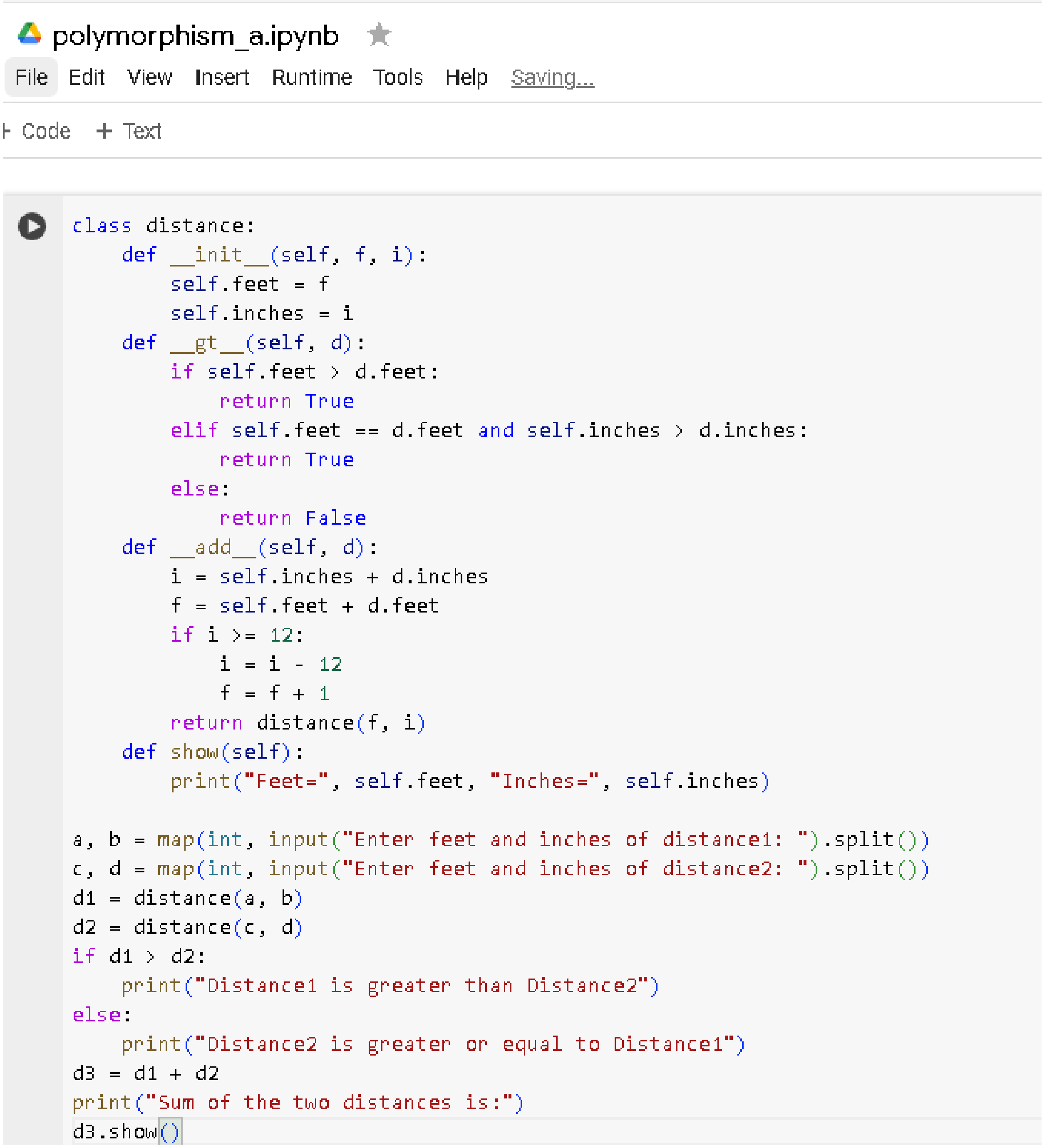
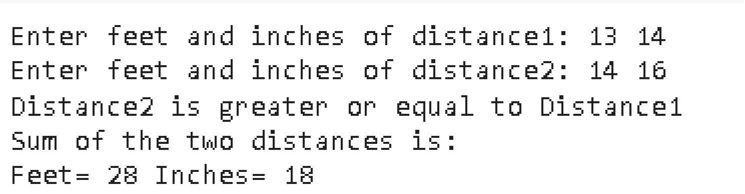
Activity Name #8- Polymorphism	
Group 3	4/29/2024
CPE 009A/CPE 12S6	Dr. Maria Rizette Sayo

5.a_3



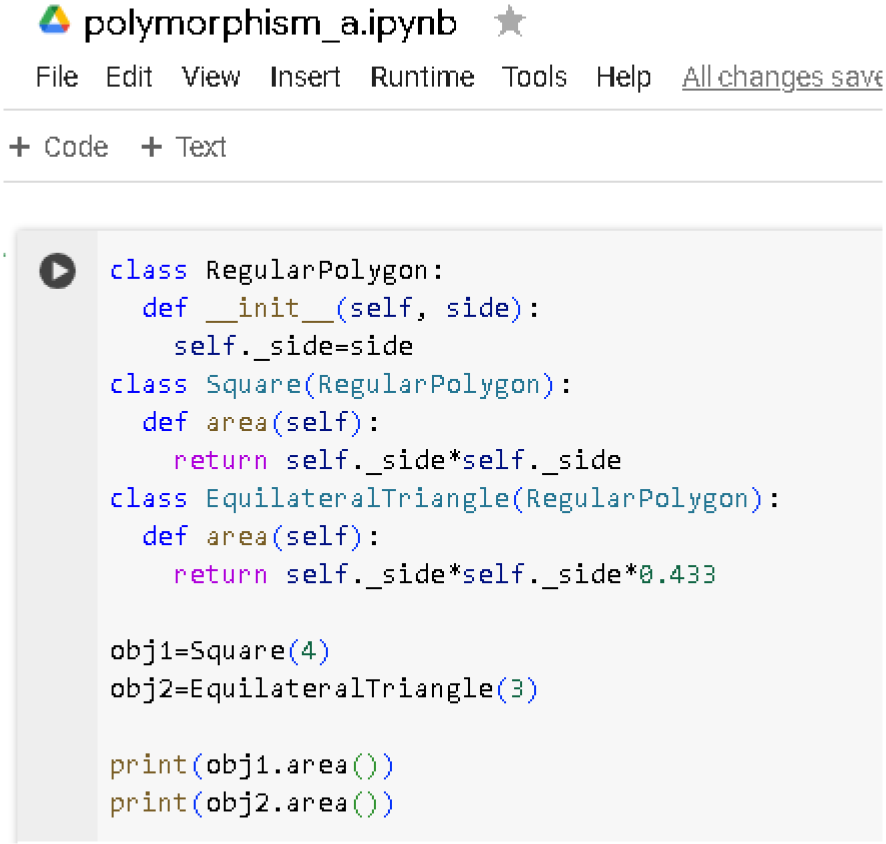
```
class distance:
    def __init__(self, f, i):
        self.feet = f
        self.inches = i
    def __gt__(self, d):
        if self.feet > d.feet:
            return True
        elif self.feet == d.feet and self.inches > d.inches:
            return True
        else:
            return False
    def __add__(self, d):
        i = self.inches + d.inches
        f = self.feet + d.feet
        if i >= 12:
            i = i - 12
            f = f + 1
        return distance(f, i)
    def show(self):
        print("Feet=", self.feet, "Inches=", self.inches)

a, b = map(int, input("Enter feet and inches of distance1: ").split())
c, d = map(int, input("Enter feet and inches of distance2: ").split())
d1 = distance(a, b)
d2 = distance(c, d)
if d1 > d2:
    print("Distance1 is greater than Distance2")
else:
    print("Distance2 is greater or equal to Distance1")
d3 = d1 + d2
print("Sum of the two distances is:")
d3.show()
```

5.a_4:



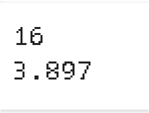
```
Enter feet and inches of distance1: 13 14
Enter feet and inches of distance2: 14 16
Distance2 is greater or equal to Distance1
Sum of the two distances is:
Feet= 28 Inches= 18
```

5.b_2:



```
class RegularPolygon:
  def __init__(self, side):
    self._side=side
class Square(RegularPolygon):
  def area(self):
    return self._side*self._side
class EquilateralTriangle(RegularPolygon):
  def area(self):
    return self._side*self._side*0.433

obj1=Square(4)
obj2=EquilateralTriangle(3)

print(obj1.area())
print(obj2.area())
```

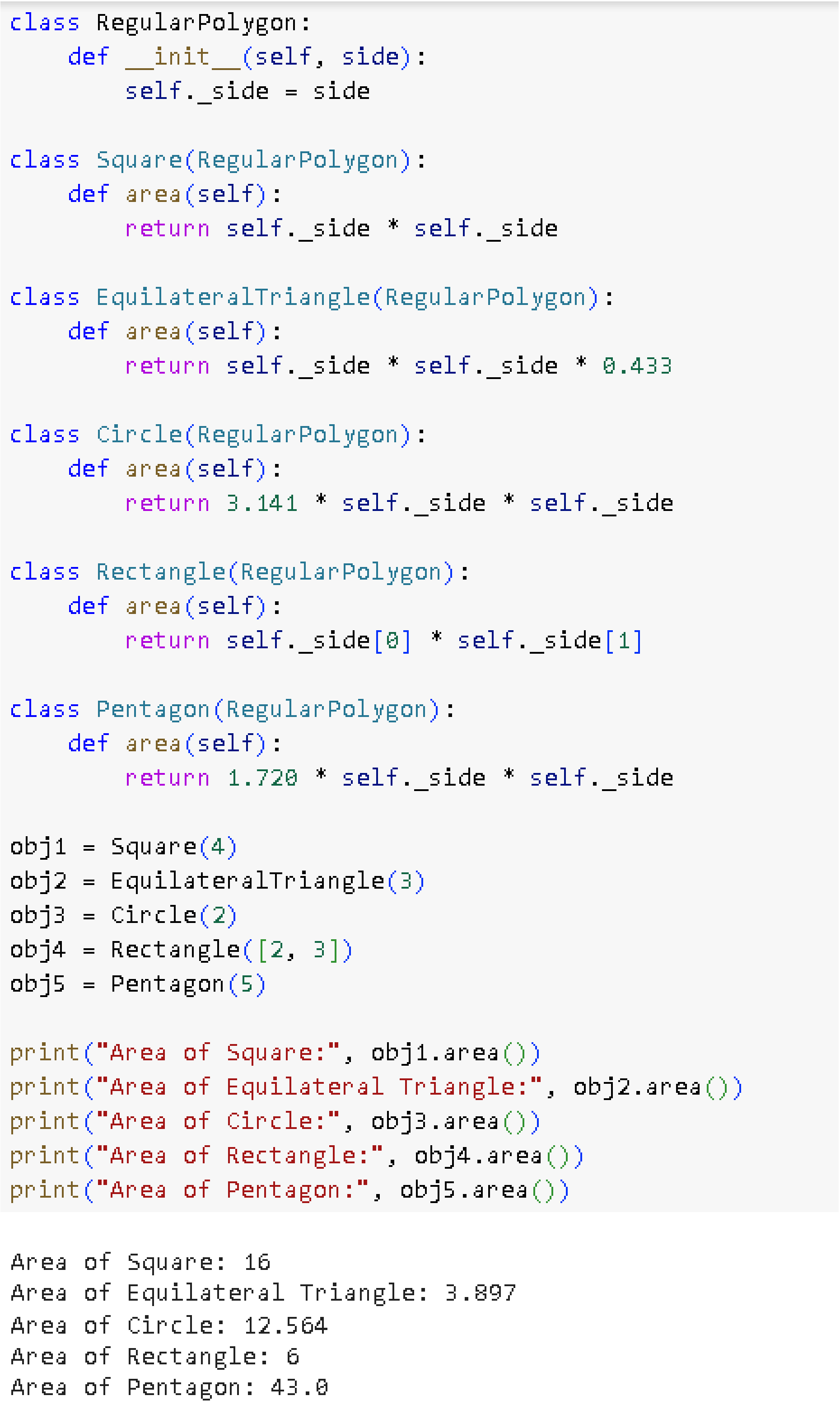
5.b_3:



```
16
3.897
```

5.b_4:

6.



```
class RegularPolygon:
    def __init__(self, side):
        self._side = side

class Square(RegularPolygon):
    def area(self):
        return self._side * self._side

class EquilateralTriangle(RegularPolygon):
    def area(self):
        return self._side * self._side * 0.433

class Circle(RegularPolygon):
    def area(self):
        return 3.141 * self._side * self._side

class Rectangle(RegularPolygon):
    def area(self):
        return self._side[0] * self._side[1]

class Pentagon(RegularPolygon):
    def area(self):
        return 1.720 * self._side * self._side

obj1 = Square(4)
obj2 = EquilateralTriangle(3)
obj3 = Circle(2)
obj4 = Rectangle([2, 3])
obj5 = Pentagon(5)

print("Area of Square:", obj1.area())
print("Area of Equilateral Triangle:", obj2.area())
print("Area of Circle:", obj3.area())
print("Area of Rectangle:", obj4.area())
print("Area of Pentagon:", obj5.area())
```

```
Area of Square: 16
Area of Equilateral Triangle: 3.897
Area of Circle: 12.564
Area of Rectangle: 6
Area of Pentagon: 43.0
```

Questions:

1.Why is Polymorphism important?:

Because it allows objects to be used interchangeably, making code more flexible and reusable. It enables different classes to be treated as instances of a common superclass, simplifying code and promoting code reusability.

2. Explain the advantages and disadvantages of using applying Polymorphism in an Object-Oriented Program.

The advantages do it apply are:

- Code Reusability: Polymorphism allows for the reuse of code by enabling objects of different classes to be treated as instances of a common superclass.
- Flexibility: It makes code more flexible and adaptable to changes, as new classes can be added without modifying existing code.
- Readability: Polymorphism can make code more readable and easier to understand by allowing objects to be used interchangeably.

While the disadvantages are:

- Complexity: Implementing polymorphism can sometimes make the code more complex and harder to debug.
- Performance Overhead: Dynamic method resolution during runtime can introduce a performance overhead compared to static method resolution.
- Misuse: If not used properly, polymorphism can lead to confusion and errors in the program.

3. What maybe the advantage and disadvantage of the program we wrote to read and write csv and json files?

Advantage of the program for reading and writing CSV and JSON files; Versatility: The program can handle different data formats, making it versatile for various data processing tasks. While as for Disadvantage of the program for reading and writing CSV and JSON files; Complexity: Handling multiple file formats can introduce complexity and potential errors in the program.

4. What maybe considered if Polymorphism is to be implemented in an Object-Oriented Program?

- Class Hierarchy: Ensure a clear and logical class hierarchy to enable meaningful polymorphic behavior.
- Method Signatures: Define common method signatures across related classes to facilitate polymorphism.
- Dynamic Binding: Understand how dynamic binding works in the programming language to leverage polymorphism effectively.
- Testing: Test different scenarios to verify that polymorphic behavior behaves as expected across different objects.

5. How do you think Polymorphism is used in an actual programs that we use today?
We use polymorphism for graphical user interface (GUI) where different types of buttons or controls can all respond to the same events.

Conclusion:
The group learned that Polymorphism in GUIs and calculators promotes code reusability, simplifies implementation, fosters extensibility, enhances user interaction, and encourages modular design.