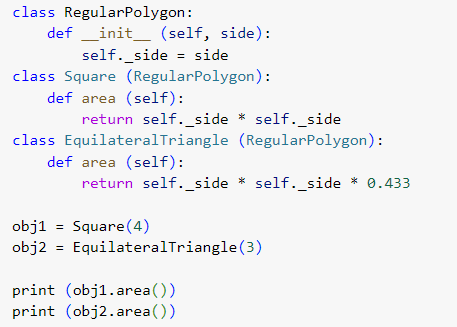
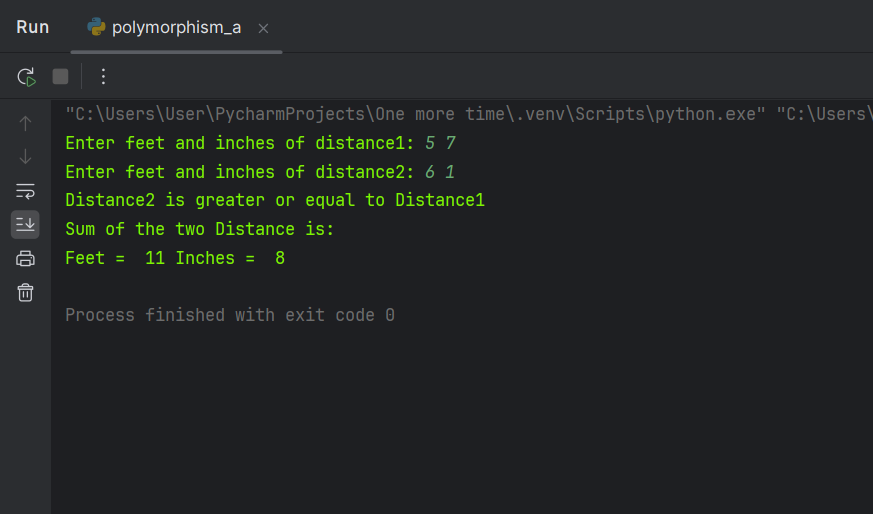
|  |  |
| --- | --- |
| **Laboratory Activity No. 7** | |
| **Polymorphism** | |
| **Course Code:** CPE103 | **Program:** BSCPE |
| **Course Title:** Object-Oriented Programming | **Date Performed:** March 16, 2025 |
| **Section:** 1 - A | **Date Submitted:** March 23, 2025 |
| **Name:** Hermosura, Leigh B. | **Instructor:** Maria Rizette M. Sayo |
| **1. Objective(s):** | |
| This activity aims to familiarize students with the concepts of Polymorphism in Object-Oriented Programming | |
| **2. Intended Learning Outcomes (ILOs):** | |
| The students should be able to:   * 1. Identify the use of Polymorphism in Object-Oriented Programming   2. Implement an Object-Oriented Program that applies Polymorphism | |
| **3. Discussion:** | |
| Polymorphism is a core principle of Object-Oriented that is also called “method overriding”. Simply stated the principles says  that a method can be redefined to have a different behavior in different derived classees.  For an example, consider a base file reader/writer class then three derived classes Text file reader/writer, CSV file reader/ writer, and JSON file reader/writer. The base file reader/writer class has the methods: read(filepath=””) , write(filepath=””). The three derived classes (classes that would inherit from the base class) should have behave differently when their read, write methods are invoked.  Operator Overloading:  Operator overloading is an important concept in object oriented programming. It is a type of polymorphism in which a user defined meaning can be given to an operator in addition to the predefined meaning for the operator.  Operator overloading allow us to redefine the way operator works for user-defined types such as objects. It cannot be used for built-in types such as int, float, char etc., For example, '+' operator can be overloaded to perform addition of two objects of distance class.  Python provides some special function or magic function that is automatically invoked when it is associated with that particular operator. For example, when we use + operator on objects, the magic  method add () is automatically invoked in which the meaning/operation for + operator is defined for user defined objects. | |
| **4. Materials and Equipment:** | |
| Windows Operating System Google Colab | |
|  | |

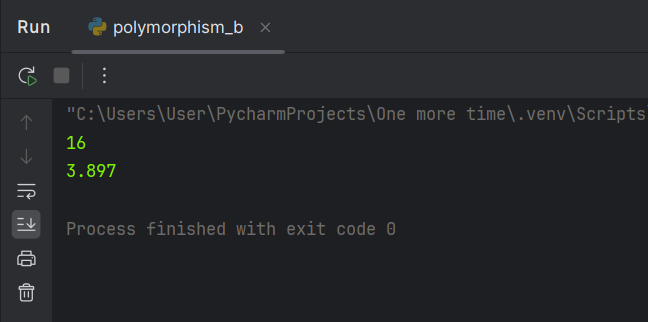
|  |
| --- |
| **5. Procedure:** |
| **Creating the Classes**   1. Create a folder named oopfa1<lastname>\_lab8 2. Open your IDE in that folder. 3. Create the base polymorphism\_a.ipynb file and Class using the code below:   Coding:  # distance is a class. Distance is measured in terms of feet and inches class distance:  def init (self, f,i):  self.feet=f self.inches=i  # overloading of binary operator > to compare two distances 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)  # overloading of binary operator + to add two distances 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)  # displaying the distance def show(self):  print("Feet= ", self.feet, "Inches= ",self.inches)  a,b= (input("Enter feet and inches of distance1: ")).split() a,b =[int(a),int(b)]  c,d= (input("Enter feet and inches of distance2: ")).split() c,d =[int(c),int(d)]  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 Distance is:") d3.show() |

4. Screenshot of the program output:





# Testing and Observing Polymorphism

1. Create a code that displays the program below:
2. Save the program as polymorphism\_b.ipynb and paste the screenshot below:
3. Run the program and observe the output.
4. Observation:

My observation is that the parent class, which is RegularPolygon, stores the value for the polygon side while the subclasses Square and EquilateralTriangle calls the parent class RegularPolygon to gain access to the value for the side and use it for computing the area for the specific type of polygon

# 6. Supplementary Activity:

In the above program of a Regular polygon, add three more shapes and solve for their area using each proper formula. Take a screenshot of each output and describe each by typing your proper labeling.

|  |
| --- |
| **Questions**   1. Why is Polymorphism important?   It’s important because it enables code reuse, eliminating the need to manually write redundant code. It is also structured, well-organized, and flexible, allowing for inheritance and repurposing of code.   1. Explain the advantages and disadvantages of using applying Polymorphism in an Object-Oriented Program.   Its advantages include reusability of code, better organization and flexibility while disadvantages include code complexity as it introduces new methods and code fragility in which minor changes can lead to unwanted problems.   1. What maybe the advantage and disadvantage of the program we wrote to read and write csv and json files?   CSV is more simpler and only uses a single line, though it lacks hierarchy unlike in JSON where code is more readable but more complex compared to CSV.   1. What maybe considered if Polymorphism is to be implemented in an Object-Oriented Program?   We should consider operator overloading, in which we can utilize operators like +, -, \*, /, == etc. to define the behavior of objects and classes. It makes the code more readable but does the opposite if used incorrectly.   1. How do you think Polymorphism is used in an actual programs that we use today?   Polymorphism is used in a variety of ways from game development, user interface, and even in payment systems for financial transactions. Polymorphism allows for reusable codes and improves code flexibility. |
| **7. Conclusion:** |
| The concepts discussed highlight the importance of structured and flexible coding practices that enhance code reuse and organization, such as inheritance and polymorphism. While CSV is simple and efficient, it lacks the hierarchy and readability of JSON, which, though more complex, offers greater flexibility. Operator overloading can make code more intuitive when used properly, but it can also reduce readability if misapplied. Polymorphism, by enabling the reuse of code in diverse applications like game development and financial systems, ultimately improves flexibility, though it may introduce complexity. Proper use of these programming concepts can lead to more efficient, readable, and adaptable code. |
| **8. Assessment Rubric:** |

For the supplementary activity, please refer to this link <https://github.com/Leigh-Hermosura/CPE-103-OOP-1-A/blob/main/Laboratory-No.7/oopfa1_Hermosura_lab7/supplementaryActivity.py>