Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

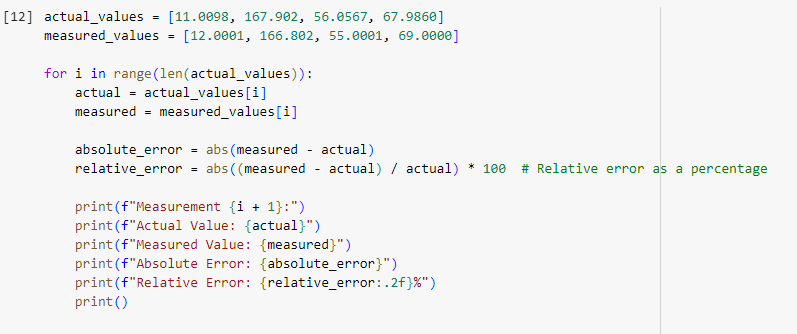
**02**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | **Write a Python program that calculates the absolute as well as relative error present in the following measurements: Actual values: [11.0098, 167.902, 56.0567, 67.9860] Measured values: [12.0001, 166.802, 55.0001, 69.0000]** |
| 2 | **Write a Python program the find the propagated error in the area and perimeter found for the following measurements: -  SQUARE (length : 45.09 cm ; uncertainty : 0.01 cm) - CIRCLE (radius : 34.90 cm ; uncertainty : 0.05 cm) - TRIANGLE (side1 : 70.9 m ; uncertainty : 0.23 m, side2 : 89.07cm ; uncertainty : 0.07 m, base : 76.07cm ; uncertainty : 0.04 m, height : 100.07cm ; uncertainty : 0.05 m) - TRAPEZIUM(side1 : 670.9 m ; uncertainty : 0.53 m (parallel one), side2 : 849.07cm ; uncertainty : 0.27 m (parallel one) side3 : 376.07cm ; uncertainty : 0.74 m, side4 : 716.07cm ; uncertainty : 0.14 m, height : 231.07cm ; uncertainty : 0.25 m)** |
| 3 | **Write a Python program that calculates the square root of following numbers using both the math.sqrt function (which uses floating-point arithmetic) and a custom square root function that uses integer arithmetic. Then, find and compare the results to observe the rounding error. (56.90, 100.45, 67.90, 25.67, 56.67)** |
| 4 | **Write a python program which finds the value of PI (π) using Taylor series, and then find the truncating error occurred due to the use of finite number of terms. HINT: arctan (1) = π/4, and formula for finding Tylor series for arctan is: 𝐚𝐫𝐜𝐭𝐚𝐧(𝐱) = ∑ (−1) 𝑛𝑧 2𝑛+1 2𝑛+1 ; ∞ 𝑛=0 |z| ≤ 1, z ≠ i, i** |

**Task 1: Write a Python program that calculates the absolute as well as relative error present in the following measurements: Actual values: [11.0098, 167.902, 56.0567, 67.9860] Measured values: [12.0001, 166.802, 55.0001, 69.0000]**

**Solution:**

**  
A screenshot of a computer

Description automatically generated**

**Task 2: Write a Python program the find the propagated error in the area and perimeter found for the following measurements:  
  
 - SQUARE (length : 45.09 cm ; uncertainty : 0.01 cm)   
- CIRCLE (radius : 34.90 cm ; uncertainty : 0.05 cm)   
- TRIANGLE (side1 : 70.9 m ; uncertainty : 0.23 m, side2 : 89.07cm ; uncertainty : 0.07 m, base : 76.07cm ; uncertainty : 0.04 m, height : 100.07cm ; uncertainty : 0.05 m)  
-TRAPEZIUM(side1 : 670.9 m ; uncertainty : 0.53 m (parallel one), side2 : 849.07cm ; uncertainty : 0.27 m (parallel one) side3 : 376.07cm ; uncertainty : 0.74 m, side4 : 716.07cm ; uncertainty : 0.14 m, height : 231.07cm ; uncertainty : 0.25 m)**

**Solution:**import math

def propagated\_error(value, uncertainty):

    return value \* uncertainty

# Square

length = 45.09

uncertainty\_length = 0.01

perimeter\_square = 4 \* length

area\_square = length \* length

propagated\_error\_square\_perimeter = propagated\_error(4 \* length, uncertainty\_length)

propagated\_error\_square\_area = propagated\_error(2 \* length, uncertainty\_length)

# Circle

radius = 34.90

uncertainty\_radius = 0.05

perimeter\_circle = 2 \* math.pi \* radius

area\_circle = math.pi \* radius\*\*2

propagated\_error\_circle\_perimeter = propagated\_error(2 \* math.pi \* radius, uncertainty\_radius)

propagated\_error\_circle\_area = propagated\_error(2 \* math.pi \* radius\*\*2, uncertainty\_radius)

# Triangle

side1 = 70.9

side2 = 89.07

base = 76.07

height = 100.07

uncertainty\_side1 = 0.23

uncertainty\_side2 = 0.07

uncertainty\_base = 0.04

uncertainty\_height = 0.05

perimeter\_triangle = side1 + side2 + base

area\_triangle = (base \* height) / 2

propagated\_error\_triangle\_perimeter = propagated\_error(side1 + side2 + base, uncertainty\_side1 + uncertainty\_side2 + uncertainty\_base)

propagated\_error\_triangle\_area = propagated\_error((base \* height) / 2, (uncertainty\_base \* height) + (uncertainty\_height \* base))

# Trapezium

side1 = 670.9

side2 = 849.07

side3 = 376.07

side4 = 716.07

height = 231.07

uncertainty\_side1 = 0.53

uncertainty\_side2 = 0.27

uncertainty\_side3 = 0.74

uncertainty\_side4 = 0.14

uncertainty\_height = 0.25

perimeter\_trapezium = side1 + side2 + side3 + side4

area\_trapezium = ((side1 + side2) / 2) \* height

propagated\_error\_trapezium\_perimeter = propagated\_error(side1 + side2 + side3 + side4, uncertainty\_side1 + uncertainty\_side2 + uncertainty\_side3 + uncertainty\_side4)

propagated\_error\_trapezium\_area = propagated\_error(((side1 + side2) / 2) \* height, (0.5 \* uncertainty\_side1 \* height) + (0.5 \* uncertainty\_height \* (side1 + side2)))

# Display results

print("Square:")

print(f"Perimeter: {perimeter\_square} cm, Propagated Error: {propagated\_error\_square\_perimeter} cm")

print(f"Area: {area\_square} cm^2, Propagated Error: {propagated\_error\_square\_area} cm^2\n")

print("Circle:")

print(f"Perimeter: {perimeter\_circle} cm, Propagated Error: {propagated\_error\_circle\_perimeter} cm")

print(f"Area: {area\_circle} cm^2, Propagated Error: {propagated\_error\_circle\_area} cm^2\n")

print("Triangle:")

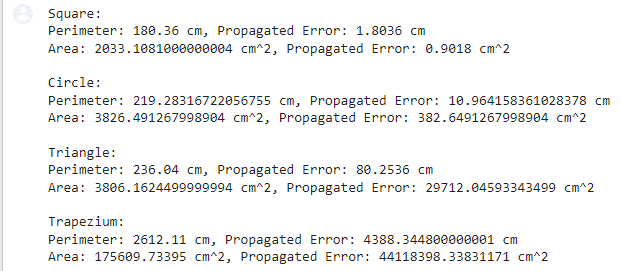
print(f"Perimeter: {perimeter\_triangle} cm, Propagated Error: {propagated\_error\_triangle\_perimeter} cm")

print(f"Area: {area\_triangle} cm^2, Propagated Error: {propagated\_error\_triangle\_area} cm^2\n")

print("Trapezium:")

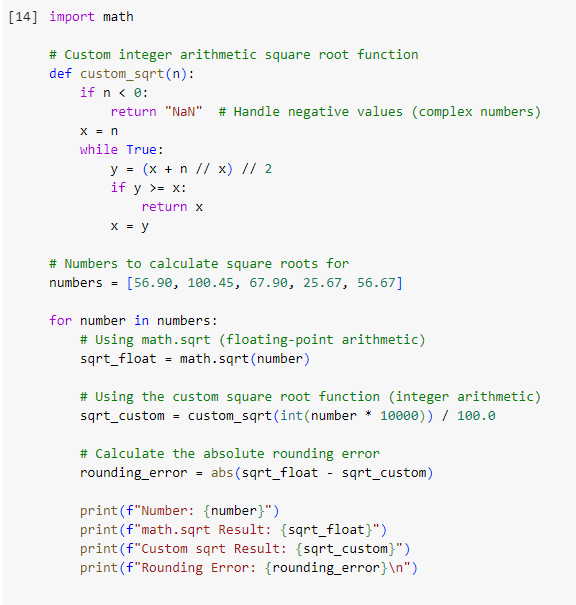
print(f"Perimeter: {perimeter\_trapezium} cm, Propagated Error: {propagated\_error\_trapezium\_perimeter} cm")

print(f"Area: {area\_trapezium} cm^2, Propagated Error: {propagated\_error\_trapezium\_area} cm^2")

****

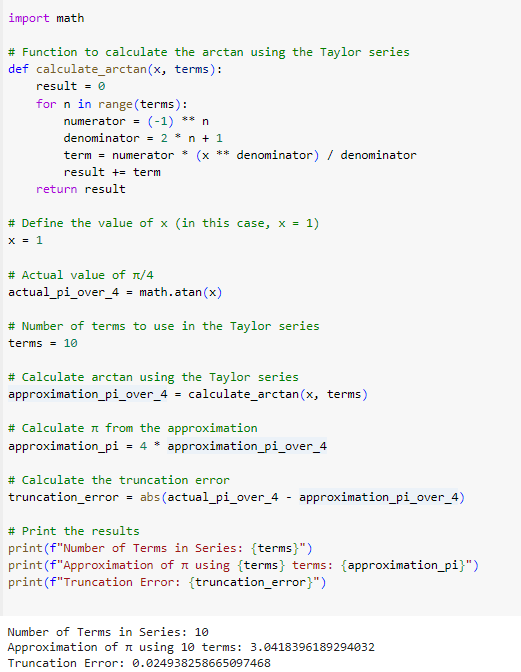
**Task 3: Write a Python program that calculates the square root of following numbers using both the math.sqrt function (which uses floating-point arithmetic) and a custom square root function that uses integer arithmetic. Then, find and compare the results to observe the rounding error.  
(56.90, 100.45, 67.90, 25.67, 56.67)**

**Solution**

****

**A screenshot of a computer

Description automatically generated**

**Task 4: Write a python program which finds the value of PI (π) using Taylor series, and then find the truncating error occurred due to the use of finite number of terms. HINT: arctan (1) = π/4, and formula for finding Tylor series for arctan is: 𝐚𝐫𝐜𝐭𝐚𝐧(𝐱) = ∑ (−1) 𝑛𝑧 2𝑛+1 2𝑛+1 ; ∞ 𝑛=0 |z| ≤ 1, z ≠ i, i  
  
**