Assignment 6

|  |  |  |
| --- | --- | --- |
| **6** | | |
| **Aim:** Create a custom Python module that contains two functions: calculate\_area for calculating the area of different shapes (circle, rectangle, triangle) and is\_prime for checking if a number is prime, then import and use this module in your main script to perform calculations. Import and use this module in your main script.  Import the datetime module and use it to display the current date and time, as well as format it in a readable way. | | |
| **Code:**  import datetime  import math  *# Basic arithmetic functions*  def add(a, b):  return a + b  def is\_even(num):  return num % 2 == 0  *# Functions for shape operations and prime check*  def calculate\_area(shape, \*dimensions):  *"""*  *Calculate area for:*  *- circle: dimensions = (radius,)*  *- rectangle: dimensions = (length, width)*  *- triangle: dimensions = (base, height)*  *"""*  if shape.lower() == 'circle':  if len(dimensions) != 1:  raise ValueError("Circle area requires 1 dimension (radius)")  radius = dimensions[0]  return math.pi \* radius \*\* 2  elif shape.lower() == 'rectangle':  if len(dimensions) != 2:  raise ValueError("Rectangle area requires 2 dimensions (length, width)")  length, width = dimensions  return length \* width  elif shape.lower() == 'triangle':  if len(dimensions) != 2:  raise ValueError("Triangle area requires 2 dimensions (base, height)")  base, height = dimensions  return 0.5 \* base \* height  else:  raise ValueError("Unsupported shape type")  def is\_prime(n):  if n <= 1:  return False  if n <= 3:  return True  if n % 2 == 0 or n % 3 == 0:  return False  i = 5  while i \* i <= n:  if n % i == 0 or n % (i + 2) == 0:  return False  i += 6  return True  def main():  *# Print current date and time in a readable format.*  now = datetime.datetime.now()  formatted\_date = now.strftime("%A, %d %B %Y %I:%M:%S %p")  print(f"Current Date and Time: {formatted\_date}\n")  *# Use basic arithmetic functions*  a, b = 5, 3  result = add(a, b)  even\_status = "even" if is\_even(result) else "odd"  print(f"Addition of {a} and {b} is {result}, which is {even\_status}.\n")  *# Shape area calculations*  try:  circle\_area = calculate\_area("circle", 4)  rectangle\_area = calculate\_area("rectangle", 5, 6)  triangle\_area = calculate\_area("triangle", 7, 8)  print(f"Area of circle with radius 4: {circle\_area:.2f}")  print(f"Area of rectangle 5x6: {rectangle\_area:.2f}")  print(f"Area of triangle with base 7 and height 8: {triangle\_area:.2f}")  except ValueError as e:  print(e)  *# Prime check*  number = 29  prime\_status = "prime" if is\_prime(number) else "not prime"  print(f"\nNumber {number} is {prime\_status}.")  if \_\_name\_\_ == "\_\_main\_\_":  main()  **Output Screenshot:** | | |
| **Conclusion/Summary:**  This assignment demonstrates several core Python programming concepts:  Function definition and parameters: Created various functions with different parameter types including standard parameters (add, is\_even), and variable arguments (calculate\_area with \*dimensions).  Conditional logic: Implemented decision-making through if-elif-else statements in the calculate\_area and is\_prime functions.  Error handling: Used raise ValueError to handle invalid inputs in the calculate\_area function.  Mathematical operations: Leveraged the math library for constants (π) and implemented various calculations.  Date and time manipulation: Used the datetime module to format and display the current date and time.  String formatting: Applied f-strings for clean output and formatted numerical results with precision specifiers.  The program successfully performs arithmetic operations, geometric calculations, and primality testing while providing meaningful output to the user. The modular design with separate functions for each responsibility makes the code maintainable and reusable. | | |
| **Student Signature & Date** | **Marks:** | **Evaluator Signature & Date** |