WEEK-5

1. **I) Write a python program that defines a matrix and prints**

```
#python program that defines a matrix and prints
from numpy import *
#accept number of rows and columns into R, C
R = int(input("Enter the number of rows:"))
C = int(input("Enter the number of columns:"))
#accept matrix elements as list
# Initialize matrix
m_list = []
print("Enter the entries row wise:")

# For user input
for i in range(R):          # A for loop for row entries
   a =[]
   for j in range(C):      # A for loop for column entries
        a.append(int(input()))
   m_list.append(a)
x=reshape(matrix(m_list),(R,C))
print("The Matrix is :")
```

1.II) Write a python program to perform addition of two square matrices

```
#To perform addition of two square matrices
import sys
from numpy import*
r1, c1 = [int(a) for a in input("First matrix rows, cols: ").split()]
r2, c2 = [int(a) for a in input("Second matrix rows, cols: ").split()]
if c1!=r2:
   print('Multiplication is not possible')
   sys.exit()
#accept First matrix elements as list
m_list1 = []
print("Enter the entries row wise:")
# For user input
for i in range(r1):          # A for loop for row entries
   for j in range(c1):       # A for loop for column entries
       m_list1.append(int(input()))
matrix1=reshape(matrix(m_list1),(r1,c1))
print("The First Matrix is :")
print(matrix1)
#accept Second matrix elements as list
m_list2 = []
print("Enter the entries row wise:")
# For user input
for i in range(r2):          # A for loop for row entries
   for j in range(c2):       # A for loop for column entries
       m_list2.append(int(input()))
matrix2=reshape(matrix(m_list2),(r2,c2))
print("The Second Matrix is :")
print(matrix2)
#Addition of two matrices
print('The Addition of Two matrix:')
add_matrix = matrix1 + matrix2
print(add_matrix)
```

1.III) Write a python program to perform multiplication of two square matrices

```
#To perform multiplication of two square matrices
import sys
from numpy import*
r1, c1 = [int(a) for a in input("First matrix rows, cols: ").split()]
r2, c2 = [int(a) for a in input("Second matrix rows, cols: ").split()]
if c1!=r2:
    print('Multiplication is not possible')
    sys.exit()
#accept First matrix elements as list
m_list1 = []
print("Enter the entries row wise:")
# For user input
for i in range(r1):          # A for loop for row entries
    for j in range(c1):      # A for loop for column entries
        m_list1.append(int(input()))
matrix1=reshape(matrix(m_list1),(r1,c1))
print("The First Matrix is :")
print(matrix1)
#accept Second matrix elements as list
m_list2 = []
print("Enter the entries row wise:")
# For user input
for i in range(r2):          # A for loop for row entries
    for j in range(c2):      # A for loop for column entries
        m_list2.append(int(input()))
matrix2=reshape(matrix(m_list2),(r2,c2))
print("The Second Matrix is :")
print(matrix2)
#Multiplication of two matrices
print('The Addition of Two matrix:')
mul_matrix = matrix1 * matrix2
print(mul_matrix)
```

2. **How do you make a module? Give an example of construction of a module using different geometrical shapes and operations on them as its functions.**

Module:

In Python, **Modules** are simply files with the ".py" extension containing Python code that can be imported inside another Python Program.

Geometry.py

```
# a module using different geometrical shapes and operations on them as its functions.
def geometrical_shapes(name):

  # converting all characters into lower cases
  name = name.lower()

  # check for the conditions
  if name == "rectangle":
    l = int(input("Enter rectangle's length: "))
    b = int(input("Enter rectangle's breadth: "))

    # calculate area of rectangle
    rect_area = l * b
    print("The area of rectangle is. ",rect_area)

  elif name == "square":
    s = int(input("Enter square's side length: "))

    # calculate area of square
    sqt_area = s * s
    print("The area of square is.",sqt_area)

  elif name == "triangle":
    h = int(input("Enter triangle's height length: "))
    b = int(input("Enter triangle's breadth length: "))

    # calculate area of triangle
    tri_area = 0.5 * b * h
    print("The area of triangle is ",tri_area)

  elif name == "circle":
    r = int(input("Enter circle's radius length: "))
    pi = 3.14

    # calculate area of circle
    circ_area = pi * r * r
    print(f"The area of circle is.",circ_area)

  elif name == 'parallelogram':
    b = int(input("Enter parallelogram's base length: "))
    h = int(input("Enter parallelogram's height length: "))

    # calculate area of parallelogram
    para_area = b * h
    print("The area of parallelogram is.",para_area)
  else:
    print("Sorry! This shape is not available")
```

main.py

```
from Geometry import *
# driver code
if __name__ == "__main__" :

  print("Calculate Shape Area")
  shape_name = input("Enter the name of shape whose area you want to find: ")

  # function calling
  geometrical_shapes(shape_name)
```

3. **Use the structure of exception handling all general purpose exceptions.**

```
# Use the structure of exception handling all general purpose exceptions.

from operator import mul, truediv

def calculate(operator, operand1, operand2):

    return operator(operand1, operand2)

try:

    first = float(input("What is your first number? "))

    second = float(input("What is your second number? "))

    operation = input("Enter either * or /: ")

    if operation == "*":

        answer = calculate(mul, first, second)

    elif operation == "/":

        answer = calculate(truediv, first, second)

    else:

        raise RuntimeError(f"'{operation}' is an unsupported operation")

except (RuntimeError, ValueError, ZeroDivisionError) as error:

    print(f"A {type(error).__name__} has occurred")

    match error:

        case RuntimeError():

            print(f"You have entered an invalid symbol: {error}")

        case ValueError():

            print(f"You have not entered a number: {error}")

        case ZeroDivisionError():

            print(f"You can't divide by zero: {error}")

else:

    print(f"{first} {operation} {second} = {answer}")
```