TASK 9 : Implement Exception and exception Handling in python
(8/10/25)

AIM : To write a Python Program that handles division by zero error using exception handing.

Algorithm:

```
1. start
2. Accept numerator and denominator from user
3. use try block to Perform division
4. if denominator is zero, catch Zerodivison Error in except block
5. Display appropriate Error message.
6. End program
```

Sample I/O

```
Enter numerator : 10
Enter denominator : 2
 Result : 5.0

Enter numerator : 10
Enter denominator : 0
~~Ente~~ Error: division by zero is not allowed !
```

P

Program

```
try:
  num1 = int(input("Enter numerator:"))
  num2 = int(input("Enter denominator:"))
    result = num1/num2
      Print("Result:", result)
    Except zero division Error:
     Print("Error: Division by zero is not allowed!")
```

TASK 9.2 Handling Multiple exceptions

AIM: To write Python Program that demonstrates handling of multiple exceptions such as invalid input and unexpected errors

Algorithm

1. Start
2. Accept a number from user.
3. use try block to calculate square of number
4. if input is not a number, handle Value Error.
5. if any other error occurs, handle it using a general exception
6. Display result
7. End

Sample I/O

```
Enter a number: 6
Square: 36

Enter a number: hello
Error: invalid input, Please enter a number!
```

Program

```
try :
    num = int (input ("Enter a number:"))
    Print ("square", num **2)
except value Error:
    Print ("Error: invalid input, Please enter a number!")
except Exception as e:
    Print ("unexpected error:",e)
```

TASK ~~8~~9.3 using finally Block

AIM : To write a python program that demonstrates the use of the finally block in exception handling.

Algorithm

1. Start the Program
2. Try to open and read from file
3. if the field is not found, handle the File Not found Error
4. use the finally block to Print a completion message
5. End

Sample I/O:

Sample.txt contains: "Hello python")

```
Hello python
    Execution Completed (finally block runs always).

Error: File not found!
    Execution Completed (finaly block runs always).
```

Program :

```
try:
    num : int (input ("Enter a number: "))
    Print ("square:", num **2)
except value Error:
        Print ("Error: Invalid input, Please enter a number!")
except Exception as e:
    Print ("unexpected error:", e)
```

✓

Program :

```
Class Negative NumberError( Exception):
    Pass
try :
  num = int (input ("Enter a Positive number:"))
  if num < 0;
     raise NegativeNumberError ("Negative number entered!")
  Print ("you Entered:", num)
  Except NegativeNumberError as e:
  Print ("Error:", e)
```

TASK 9.4 user - Defined Exception

AIM : To write a Python program that demonstrate user - defined exceptions

Algorithm :

1. start
2. Define a Custom exception class
3. Accept a number from user
4. if number is negative, raise user - defined exception.
5. Catch the exception in except block and display the error
6. If no error, Print number entered
7. End

Sample I/O

```
Enter a Positive number: 15
You entered: 15

Enter a Positive number: 8
   Error: Negative number entered!
```

[illegible]

VEL TECH	
EX No.	9
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	
RECORD (5)	5
TOTAL (20)	5
SIGN WITH DATE	20

Result : Thus the Exception handling Python has been verified.