

TASK: 9
DATE: 8/10/2023

IMPLEMENT EXCEPTION OR EXCEPTION HANDLING IN PYTHON

Q-1 : DIVISION BY ZERO EXCEPTION

Ans:-

To write a python program that handles division by zero error using exception handling

Algorithm:-

- 1) Start
- 2) Accept numerator & denominator from user
- 3) Use try block to perform division.
- 4) If denominator is zero, catch ZeroDivisionError in except block.
- 5) Display the appropriate error message
- 6) End

I/O:-

Enter numerator: 10

Enter denominator: 2

Results: 5.0

Enter numerator: 10

Enter denominator: 0

Error: Division by zero is not allowed!

Program:

```
num1 = int(input("Enter numerator: "))
num2 = int(input("Enter denominator: "))
result = num1 / num2
print("Result:", result)
except ZeroDivisionError:
    print("Error: Division by zero is not allowed!")
```

9-2 Handling Multiple Exceptions:-

Ans:-

To write a python program that handles handling of multiple exceptions such as individual input & unexpected errors.

Algorithm:-

- 1) Start
- 2) Accept a number from the user
- 3) Use try block to calculate the square of the number.
- 4) If the input is not a number, handle ValueError.
- 5) If any other error occurs, handle it using a general Exception.
- 6) Display the results or error message
- 7) End

Example 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 ValueError:

```
    print("Error: Invalid input, please enter a number")
```

except Exception as e:

```
    print("Unexpected error: " e)
```

for i in range(1, 11):
 print(i * i)

O : rotation 90°

S : Rotational reading

G : 8

O : rotation 90°

O : Rotational reading

O : rotation 90°

9.3 Using Finally Block:-

Aim:-

To write a Python program that demonstrates how we use exception handling.

Algorithm:-

- 1) Start
- 2) Try to Open & Read from a file.
- 3) If the file is not found, handle the FileNotFoundError.
- 4) Use the finally block to print a completion message.
- 5) End.

Program:- Output

(sample.txt contains "Hello Python")

Hello Python

Execution completed (finally blocks runs always).

Error: Filenot found!

Execution completed (finally block runs always).

REVIEW:-

This topic deals with exception handling in Python and discussed

Program:-

```
try:  
    file = open("sample.txt", "r")  
    content = file.read()  
    print(content)  
except FileNotFoundError:  
    print("Error: File not found!")  
finally:  
    print("Execution Complete (finally block  
runs always).")
```

9.4 : User -Defined Exception

Aim:- To write a Python program that demonstrates user-defined exceptions.

ALGORITHMS:-

- 1) Start
- 2) Define a custom exception class
- 3) Accept a number from the user
- 4) If the number is negative, raise the user-defined exception.
- 5) Catch the exception in the except block & display the error msg
- 6) If no error, print the number entered
- 7) End the program.

I/O:-

Enter a positive number: 15

You entered: 15

Enter a positive number: -8

You entered negative number

RESULT:-

VELTECH	
EX NO.	9
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	qr
SIGN WITH DATE	dk

Thus To implement exception of exception handling in python has been verified and executed.

Program :-

Class NegativeNumberError (Exception).

if num < 0 then throw new NegativeNumberError ("Negative number entered")

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)