**10 (Exception Handling)**

**Q1: Write a function to compute 5/0 and use try/except to catch the exceptions.**

def catchException(x,y):

try:

print(x/y)

except Exception:

print("Zero Division error : Denominator can not be zero")

first\_number=int(input("Enter first number: "))

second\_number=int(input("Enter second number: "))

catchException(first\_number,second\_number)

Enter first number: 5

Enter second number: 0

Zero Division error : Denominator can not be zero

**Q2: Write a Python program to access the array element whose index is out of bound and handle the corresponding exception**

array=[1,3,7,5]

try:

print(array[7])

except Exception as e:

print("error occured: ",e)

error occured: list index out of range

**Q 3: Write a Python Program to handle multiple errors with one except statement**

**def catchException():**

first\_number=(input("Enter first number: "))

second\_number=(input("Enter second number: "))

try:

print(int(first\_number)/int(second\_number))

except (ZeroDivisionError,ValueError) as e:

print("Error occured: ",e)

catchException()

Enter first number: 2

Enter second number: a

Error occured: invalid literal for int() with base 10: 'a'

**Q 4: Write a Python Program to depict else clause with try-except**

def catchException():

first\_number=(input("Enter first number: "))

second\_number=(input("Enter second number: "))

try:

print(int(first\_number)/int(second\_number))

except (ZeroDivisionError,ValueError) as e:

print("Error occured: ",e)

else: #This will be executed when there is no exception

print("code execued.")

finally: #This will be executed irrespective of exception

print("End of the program.")

catchException()

Enter first number: 5

Enter second number: 0

Error occured: division by zero

End of the program.

**Q5: Write a Python Program to depict Raising Exception**

try:

x=int(input("Enter positive integer: "))

if (x<0):

raise ValueError("Entered value is not positive integer.")

else:

print("Entered value is positive integer: ",x)

except ValueError as v:

print("Error occured: ",v)

Enter positive integer: -5

Error occured: Entered value is not positive integer.

**Q6: Write a python program to create user-defined exception**

class customClass(Exception):

'''This is custom defined class'''

pass

class valueSmallError(customClass):

pass

class valueLargeError(customClass):

pass

num=20

while True:

try:

num1=int(input("Enter number: "))

if num1<num:

raise valueSmallError

elif num1>num:

raise valueLargeError

break

except valueSmallError:

print("enter larger number.")

except valueLargeError:

print("enter smaller number.")

print(f"given number is {num} and your guessing number is {num1}.Correct.")

Enter number: 25

enter smaller number.

Enter number: 18

enter larger number.

Enter number: 19

enter larger number.

Enter number: 20

given number is 20 and your guessing number is 20.Correct.

**Q7: Print one message if the try block raises a NameError and another for other errors**:

try:

if (y):

raise NameError

x=int(input("Enter integer: ")) #This is to check other errors

except NameError as n:

print("Name Error is occured: ",n)

except Exception as e:

print("This can handle remaining errors other than Name Error,because name error is called first.",e)

Name Error is occured: name 'y' is not defined

**Q8: Try to open and write to a file that is not writable:**

import os

try:

f = open("names.txt", "w") #created new file

f.write("Elif") #added some content

with open("names.txt") as f: #opened file without specifying the mode

f.write("Hi")

except Exception as e:

print("Error: ",e)

Error: not writable

**Q9: Write a python program to raise an error and stop the program if x is lower than 0.**

try:

x=float(input("Enter value greater or equal to 0: "))

if (x<0):

raise ValueError("Entered value is less than 0.")

else:

print("Entered value is : ",x)

except ValueError as v:

print("Error occured: ",v)

Enter value greater or equal to 0: -1

Error occured: Entered value is less than 0.

**Q10: Write a python program to raise a TypeError if x is not an integer:**

try:

x=int(input("Enter integer: "))

print(x)

except Exception as e:

print("TypeError has occurred. Given value is not an integer.")

Enter integer: 5.2

TypeError has occurred. Given value is not an integer.