python-programming-lab-10

March 13, 2025

Python Programming - 2301CS404

Gohel Neel

Enrollnment No. : 23010101089

Roll No. 340

Date: 03-02-2025

Lab - 10

1 Exception Handling

1.0.1 01) WAP to handle following exceptions:

1. ZeroDivisionError
2. ValueError
3. TypeError #### Note: handle them using separate except blocks and also using single except block too.

[10]:
```
try:
    a = int(input("Enter 1st number "))
    b = int(input("Enter 2nd number "))
    c = a//b
    print(c)
except (ZeroDivisionError , ValueError , TypeError ) as e:
    print(e)
```

```
Enter 1st number 10
Enter 2nd number 0
integer division or modulo by zero
```

1.0.2 02) WAP to handle following exceptions:

1. IndexError
2. KeyError

[3]:
```
l1 = [1,2,3]
d1 = {1 : 'a' , 2:'b'}
try:
```

```
    print(l1[3])
    print(d1[3])
except (IndexError ,KeyError ) as e:
    print(e)
```

```
'c'
```

1.0.3 03) WAP to handle following exceptions:

1. FileNotFoundError
2. ModuleNotFoundError

[35]:
```
try:
    fp = open("temp.txt" , "r")
except FileNotFoundError as e:
    print(e)
try:
    import neel
except ModuleNotFoundError as err:
    print(err)
```

```
No module named 'neel'
```

1.0.4 04) WAP that catches all type of exceptions in a single except block.

[14]:
```
try:
    a = int(input("Enter Number "))
    b = int(input("Enter Number "))
    c = a // b
    print(c)
except Exception as e:
    print(e)
```

```
Enter Number 10
Enter Number ru
invalid literal for int() with base 10: 'ru'
```

1.0.5 05) WAP to demonstrate else and finally block.

[16]:
```
try:
    fp = open('temp.txt' , "r")
except Exception as e:
    print(e)
else:
    print(fp.read())
    fp.close()
finally:
    print("hello from the finally block")
```

```
hello
hello from the finally block
```

1.0.6 06) Create a short program that prompts the user for a list of grades separated by commas.

1.0.7 Split the string into individual grades and use a list comprehension to convert each string to an integer.

1.0.8 You should use a try statement to inform the user when the values they entered cannot be converted.

[ ]:
```
```

1.0.9 07) WAP to create an udf divide(a,b) that handles ZeroDivisionError.

[26]:
```
class divideError(Exception):
    pass
try:
    a = int(input("Enter Number "))
    b = int(input("Enter Number "))

    if(b != 5):
        print(a // b)
    else:
        raise divideError

except divideError:
    print("error")
```

```
Enter Number 10
Enter Number 5
error
```

1.0.10 08) WAP that gets an age of a person form the user and raises ValueError with error message: "Enter Valid Age" :

If the age is less than 18.

otherwise print the age.

[30]:
```
class ageError(Exception):
    pass

try:
    age = int(input("Enter age "))
    if(age > 18):
        print(age)
    else:
```

```
        raise ageError
except ageError:
    print("Enter Valid Age")
```

```
Enter age 19
19
```

1.0.11 09) WAP to raise your custom Exception named InvalidUsernameError with the error message : “Username must be between 5 and 15 characters long”:

if the given name is having characters less than 5 or greater than 15.

otherwise print the given username.

[34]:
```
class InvalidUsernameError(Exception):
    pass

try:
    name = input("Enter your name ")
    if(len(name) > 5 and len(name) < 15):
        print(name)
    else:
        raise InvalidUsernameError
except InvalidUsernameError:
    print("Username must be between 5 and 15 characters long")
```

```
Enter your name neel
Username must be between 5 and 15 characters long
```

1.0.12 10) WAP to raise your custom Exception named NegativeNumberError with the error message : “Cannot calculate the square root of a negative number” :

if the given number is negative.

otherwise print the square root of the given number.

[40]:
```
import math
class NegativeNumberError(Exception):
    def __init__(self, msg):
        self.msg = msg
try:
    a = int(input("Enter number "))
    if(a > 0):
        ans = (math.sqrt(a))
        print(ans)
    else:
        raise NegativeNumberError("Cannot calculate the square root of a␣
 ↪negative number")
```

```
except NegativeNumberError as e:
    print(e)
```

```
Enter number 56
7.483314773547883
```

[ ]: