Chinmai Kulkarni

Python Training

Session 8

1. Value error, type error, Name error examples

**def** name():  
 **try**:  
 a+b  
 **except** NameError:  
 print(**"Please define a value for a and b!!"**)  
 **finally**:  
 print(**"Thank you"**)  
name()

**def** name():  
 **try**:  
 x = eval(input(**"Enter your age"**))  
 a = x+10  
 print(**"age after 10 years is:"**,a)  
 **except** TypeError:  
 print(**"Age cannot be added with string!!"**)  
 **finally**:  
 print(**"Thank you"**)  
print(name())

**try**:  
 number = int(input(**"Enter the number "**))  
 re = 10/number  
**except** ValueError :  
 print (**"Value is not int type"**)

1. Multiple except block – Yes multiple except works instead of a single except block.

**while**(**True**):  
 **try**:  
 number = int(input(**"Enter the number "**))  
 re = 10/number  
 **except** ValueError :  
 print (**"Value is not int type"**)  
 **except** ZeroDivisionError :  
 print (**"Please try again with a different number"**)  
 **else**:  
 print (**"result is "**,re)   
 **break**

1. Multiple try and except block – Yes but the result shows the output of both except blocks.

**while**(**True**):  
 **try**:  
 number = int(input(**"Enter the number "**))  
 **except** ValueError :  
 print (**"Value is not int type"**)  
 **try**:  
 re = 10 / number  
 **except** ZeroDivisionError :  
 print (**"Please try again with a different number. You cannot input 0"**)  
 **else**:  
 print (**"result is "**,re)  
 **break**

1. Raise in python

Raise is used to define custom errors in python. The lines after raise are never executed.

Ex:

**try**:  
 a = 10  
 b = 25  
 **if** b > 20:  
 **raise** Exception(**"Please give a value less than 20"**)  
 print(**'After raise'**)   
**except**:  
 print(**"Please enter value less than 20"**)  
**finally**:  
 print(**"Thank you!"**)

1. Can finally be used in between, top or bottom? – Yes, finally block has to be used in the middle and at the end after except block.

**try**:  
 a = 10  
 b = 20  
 **try**:  
 res = a + b  
 print(res)  
 **finally**: #finally  
 print (**"Thank you"**)  
**except** (TypeError,NameError):  
 print (**"please try again"**)

**def** name():  
 **try**:  
 a = 10  
 b = 20  
 print(a+b)  
 **except** NameError:  
 print(**"Please define a value for a and b!!"**)  
 **finally**:  
 print(**"Thank you"**)  
name()

1. Can or be used in list comprehension instead of by default and? – Yes, or can be used instead of by default and between if’s but with only one if statement.

x = [i **for** i **in** range(0,20) **if** i%2==0 **or** i%5==0]  
print(x)