UDITI NAMDEV SE-B 60004180111

IBM Python Exercise – 2 Exception Handling

1) Code:

```
num=int(input("Enter the numerator:"))
deno=int(input("Enter the denominator:"))
try:
   quo=num/deno
   print("QUOTIENT:",quo)
except ZeroDivisionError:
   print("Denominator cannot be zero")
```

Output:

```
Enter the numerator:10
Enter the denominator:2
QUOTIENT: 5.0
>>>
============= RESTART: /Users/Uditi/Desktop/Assignment 2 part1.py =============
Enter the numerator:10
Enter the denominator:0
Denominator cannot be zero
```

Explanation:

This program is used to find the quotient by taking in the numerator and denominator from the user.
Code in the try block is executed for exceptions.
Division of two numbers is not possible if the denominator is zero, thus we have used the exception “except ZeroDivisionError” clause to print the message that Denominator can’t be zero in case of a logical error of putting in the denominator as zero.

2)Code:

```
string=input("Enter some string:")
try:
   x=string[5]
   print("char at index 5 is ",x)
   print("no exception")
except IndexError as e:
   print("Exception raised",e)
else:
   print("user entered value of x is:",x)
   print("else block executed")
finally:
   x=6
   print("finally block always execute")
   print("value of x:",x)
```

Output:

```
Enter some string:Uditi
char at index 5 is  h
no exception
user entered value of x is: h
else block executed
```

```
finally block always execute
value of x: 6
>>>
============= RESTART: /Users/Uditi/Desktop/Assignment 2 part1.py =============
Enter some string:Pugs
Exception raised string index out of range
finally block always execute
value of x: 6
```

Explanation:

This program is used to check if the string entered by the user has 5 or more characters and to print the 6th character.

Here the code in the try block is always tried for exception.

If the entered string is more than 6 characters then x is allocated a value and the IndexError is not raised.then the 6th character of the string is printed.

If the entered string is less than 5 characters then the value of x is not allocated and IndexError is raised.

Finally clause is always executed no matter if the error is raised or not.

3)Code:

```
try:
   value_one=int(input("Enter First Number:"))
   value_second=int(input("Enter Second Number:"))
   print("The Result:",value_one/value_second)
except ZeroDivisionError:
   print("ZeroDivisionError:cannot divide with zero")
except:
   print("Default Except:please provide valid input only")
```

Output:

```
Enter First Number:10
Enter Second Number:5
The Result: 2.0
>>>
============= RESTART: /Users/Uditi/Desktop/Assignment 2 part1.py =============
Enter First Number:9
Enter Second Number:0
ZeroDivisionError:cannot divide with zero
>>>
============= RESTART: /Users/Uditi/Desktop/Assignment 2 part1.py =============
Enter First Number:try
Default Except:please provide valid input only
```

Explanation:

The program is used to input two numbers from the user and divide them and print the quotient.

Code in the try block is executed for exceptions.

If there are no exceptions found then the code gets executed and the result.

If the second value i.e. the denominator is 0 then the ZeroDivisionError exception is raised and hence a statement “cannot divide with zero” is printed

If there are any other exceptions for example NumberFormatException (character or string is entered instead of a number) is raised then it is led to the except block in order to provide a valid input.

4)Code:

```
try:
   a = int(input("Enter a: "))
   b = int(input("Enter b: "))
   if b== 0:
      raise ArithmeticError;
   else:
      print("a/b = ",a/b)
except ArithmeticError:
   print("The value of b can't be 0")
```

Output:

```
Enter a: 10
Enter b: 2
a/b =  5.0
>>>
============= RESTART: /Users/Uditi/Desktop/Assignment 2 part1.py =============
Enter a: 9
Enter b: 0
The value of b can't be 0
```

Explanation:

This program is used to input two numbers from the user and divide them and print the quotient.
Code in the try block is executed for exceptions.
First using the if condition the value of b is checked whether it is 0 or not.
If it is, then the ArithmeticError is raised and led to the except block printing that the value of b can’t be zero.
If it is not, then the code is run in order to print the quotient.

5)Code:

```
try:
   age = int(input("Enter the age: "))
   if age<18:
      raise ValueError;
   else:
      print("The age is valid")
except ValueError:
   print("The age is not valid")
```

Output:

```
Enter the age: 20
The age is valid
>>>
============= RESTART: /Users/Uditi/Desktop/Assignment 2 part1.py =============
Enter the age: 10
The age is not valid
>>>
============= RESTART: /Users/Uditi/Desktop/Assignment 2 part1.py =============
Enter the age: try
The age is not valid
```

Explanation:

This program is used to check whether the age entered is valid or not.(age above 18 is valid)
Code in the try block is executed for exceptions.
First the age entered is checked whether it is greater than 18 or not.
If the age is not greater than 18 then ValueError exception is raised which is further led to the except block to print that the age is not valid.
If the age is greater than 18 then the else block is executed printing that the age is valid.