Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2101CS405

Lab - 9

Exception Handling

In [ ]:
```
Name : VORA YAGNIK
Enrollment no: 23010101661
```

A

01) WAP to handle divide by zero exception.

In [1]:
```
try:
    a = 5/0
except ZeroDivisionError:
    print("Divide By Zero Error is occured")
```

```
Divide By Zero Error is occured
```

02) Write a Python program that inputs a number and generates an error message if it is not a number.

In [3]:
```
try:
    a = int(input("Enter a number : "))
except ValueError:
    print("Please Enter a Number")
```

```
Please Enter a Number
```

03) WAP to handle file not found Exception

In [6]:
```
try:
    file = open("demo.txt","r")
except FileNotFoundError:
    print("Please open existing file")
```

```
Please open existing file
```

04) WAP to handle type Exception.

In [7]:
```
try:
    a = "a" + 5
except TypeError:
    print("Type Error is Occured")
```

```
Type Error is Occured
```

05) WAP to demonstrate valueError and indexError with example.

In [8]:
```
def valueErrorDemo():
    try:
        a = int("a")
    except ValueError:
        print("ValueError is occured")

def indexErrorDemo():
    try:
        a = [1,2,3]
        b = a[5]
    except IndexError:
        print("IndexError is occured")

valueErrorDemo()
indexErrorDemo()
```

```
ValueError is occured
IndexError is occured
```

06) WAP to domonstrate else and finally block.

In [13]:
```
def demo(a, b):
    try:
        result = a / b
    except ZeroDivisionError:
        print("Division by zero")
    else:
        print("else :::: Division successful. Result:", result)
    finally:
        print("finally :::: Finally block executed")

demo(10, 2)
demo(10, 0)
```

```
else :::: Division successful. Result: 5.0
finally :::: Finally block executed
Error: Division by zero
finally :::: Finally block executed
```

07) Create a short program that prompts the user for a list of grades separated by commas. Split the string into individual grades and use a list comprehension to convert each string to an integer. You should use a try statement to inform the user when the values they entered cannot be converted.

In [23]:
```
try:
    grades = input("Enter the list of grades separated by commas: ")
    grade_list = grades.split(',')
    gradesToInt = [int(grade) for grade in grade_list]
    print("Grades:", grades)
except ValueError:
    print("Please Enter Valid format")
```

```
Please Enter Valid format
```

B

01) WAP to Raising User Generated Exception.

In [29]:
```
class MyException(Exception):
    def __init__(self,arg):
        self.arg = arg

try:
    a = int(input("Enter a positive number"))
    if a < 0:
        raise MyException("Enter positive number")
except MyException as e:
    print(e.arg)
```

```
Enter positive number
```

02) WAP to raise your custom Exception.

In [31]:
```
class MyException(Exception):
    def __init__(self,arg):
        self.arg = arg

try:
    a = int(input("Enter a positive number"))
    if a%2 == 0:
        raise MyException("Enter number is even")
    else:
        raise MyException("Entered number is odd")
```

```
except MyException as e:
    print(e.arg)
```

```
Entered number is odd
```