(https://www.darshan.ac.in/)

Python Programming - 2101CS405

Lab - 9

Name:Parmar himanshu

Roll No.:B3-341

Enrollment No.:22010101132

Exception Handling

A

01) WAP to handle divide by zero exception.

In [2]:
```
num1 = int(input("Enter 1st Number:"))
num2 = int(input("Enter 2nd Number:"))

try:
    ans = num1 / num2
    print(f"Answer:{ans}")
except ZeroDivisionError:
    print("Error: Division by zero")
```

```
Enter 1st Number:10
Enter 2nd Number:2
Answer:5.0
```

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

In [6]:
```
num = input("Enter Number:")

try:
    ans = int(num)
    print(ans)
except:
    print("Enter Valid Number")
```

```
Enter Number:2a
Enter Valid Number
```

03) WAP to handle file not found Exception

In [12]:
```
try:
    f = open("demo1.py","r")
    data = f.read()
    print(data)
except FileNotFoundError:
    print("Couldn't find file")
```

04) WAP to handle type Exception.

In [22]:
```
num = str(input("Enter Value:"))
try:
    ans = num/10
except TypeError:
    print("Invalid Datatype or Operattion")
```

```
Enter Value:abc
Invalid Datatype or Operattion
```

05) WAP to demonstrate valueError and indexError with example.

In [27]:
```
num = input("Enter Value:")

try:
    ans = int(num)
except ValueError:
    print("Value Error")

list = [12,23,4,2]
try:
    ind = int(input("Enter index:"))
    print(list[ind])
except IndexError:
    print("Index Error")
```

```
Enter Value:12
Enter index:65
Index Error
```

06) WAP to domonstrate else and finally block.

In [29]:
```
try:
    ans = 5 + "70"
except TypeError:
    print("Invalid Datatype")
else:
    print("else Block")
finally:
    print("finally Block")
```

```
Invalid Datatype
finally Block
```

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 [37]:
```
grade = input("Enter grade With comma Seprated:")
list = grade.split(",")
try:
    ans = [int(i) for i in list]
    print(ans)
except ValueError:
    print("Value Can't Converted")
```

```
Enter grade With comma Seprated:10,20,30
[10, 20, 30]
```

Type *Markdown* and LaTeX: α^2

B

01) WAP to Raising User Generated Exception.

In [50]:
```
num = input("Enter Value:")

try:
    if num.isdigit():
        print(num)
    else:
        raise TypeError
except TypeError:
    print("Invalid Datatype or Operattion")
```

```
Enter Value:ab
Invalid Datatype or Operattion
```

02) WAP to raise your custom Exception.

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

num = input("Enter Value:")
try:
    if num.isdigit():
        print(num)
    else:
        raise TypeConverteError("Type Convert Error")
except TypeConverteError as e:
    print(e.arg)
```