

## Python Programming - 2101CS405

### Lab - 9

Name :Vyas Bhagyesh Y.

Enrollment No : 23010101662

Roll NO : 23010101662

## Exception Handling

### A

01) WAP to handle divide by zero exception.

```
In [1]: n=int(input("Enter a number-1 : "))
b=int(input("Enter a number-2 : "))
try:
    c=n/b
    print(c)
except ZeroDivisionError:
    print("Zero Division Error")
```

Zero Division Error

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

```
In [2]: try:
n=int(input("Enter a number : "))
except ValueError:
    print("Enter valid number")
```

Enter valid number

### 03) WAP to handle file not found Exception

```
In [3]: try:
        f=open("demo.txt")
    except FileNotFoundError:
        print("Couldn't find file")
```

Couldn't find file

### 04) WAP to handle type Exception.

```
In [4]: try:
        "a"+1
    except TypeError:
        print("Invalid Datatype or Operation")
```

Invalid Datatype or Operation

### 05) WAP to demonstrate valueError and indexError with example.

```
In [5]: l1=[1,2,3,4,5,6,7,8]
    try:
        a=int(input("Enter number :"))
        print("Value = ",l1[a])
    except (ValueError,IndexError):
        print("Value Error & Index Error")
```

Value Error & Index Error

### 06) WAP to domonstrate else and finally block.

```
In [6]: try:
        n=int(input("Enter a number : "))
    except ValueError:
        print("Invalid Datatype")
    finally:
        print('Finally block executed')
```

Invalid Datatype

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 [2]: a = input("Enter comma separated grades : ")
    li = a.split(",")
    try:
        li1= list(i for i in li if int(i)>0 or int(i)<=0)
        print(li1)
    except ValueError:
        print("Value Can't Converted")
```

Value Can't Converted

# B

## 01) WAP to Raising User Generated Exception.

```
In [7]: class PostiveNmuberException(Exception):
        def __init__(self, arg):
            self.arg=arg
        try:
            n=int(input("Enter positive number : "))
            if(n>0):
                print(n)
            else:
                raise PostiveNmuberException("Enter Postive Number")
        except PostiveNmuberException as e:
            print(e.arg)
```

Enter Postive Number

## 02) WAP to raise your custom Exception.

```
In [8]: class OddNumberException(Exception):
        def __init__(self, arg):
            self.arg=arg
        try:
            n=int(input("Enter a number : "))
            if(n%2!=0):
                print(n)
            else:
                raise OddNumberException("Even Number Entered")
        except OddNumberException as e:
            print(e.arg)
```

Even Number Entered