



[\(https://www.darshan.ac.in/\)](https://www.darshan.ac.in/)

Python Programming - 2101CS405

Lab - 9

Name : Viral Chauhan

Enrollment : 22010101027

Roll No. : 184 Batch : A4

Exception Handling

```
In [1]: try:
        a=5
        b="3"
        c=a+b
    except TypeError:
        print("Invalid Datatype")
    else:
        print("else Block Excuted")
    finally:
        print("finally Block Excuted")
```

Invalid Datatype
finally Block Excuted

A

01) WAP to handle divide by zero exception.

```
In [1]: try:
        a=int(input("Enter a number : "))
        b=0
        c=a/b
    except ZeroDivisionError:
        print("divide by zero not possible")
```

Enter a number : 4
divide by zero not possible

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

```
In [4]: try:
        a=int(input("Enter a number : "))

    except ValueError:
        print("value error")
```

Enter a number : d
value error

03) WAP to handle file not found Exception

```
In [8]: try:
        f = open("kishan.txt", 'r')
    except FileNotFoundError:
        print("file not found")
```

file not found

04) WAP to handle type Exception.

```
In [10]: try:
        a="parth"
        b= 1
        c = b+ a
    except TypeError:
        print("type error")
```

type error

05) WAP to demonstrate valueError and indexError with example.

```
In [23]: def errorFun(a,b='as'):
        try:
            a=int(a)
            b= b[10]
        except ValueError:
            print("Value error")
        except IndexError:
            print("Index error")
```

```
errorFun('a')
errorFun('10', 'sdfg')
```

Value error

Index error

06) WAP to demonstrate else and finally block.

```
In [26]: def errorFun(a):  
    try:  
        b= 5/a  
    except ZeroDivisionError:  
        print("Zero Division Error",a)  
    else:  
        print("this is else, only when exception is not occured",a)  
    finally:  
        print("i am finally, every time ",a)  
errorFun(5)  
errorFun(0)
```

```
this is else, only when exception is not occured 5  
i am finally, every time 5  
Zero Division Error 0  
i am finally, every time 0
```

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 [28]: grades = input("enter comma separated marks : ")  
grades = grades.split(",")  
myList = []  
try :  
    myList = [int(i) for i in grades]  
except ValueError:  
    print("value can't converted")
```

```
enter comma separated marks : 5,6,p  
value can't converted
```

B

01) WAP to Raising User Generated Exception.

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

          a = int(input("Enter a positive number"))
          if(a<0):
              raise MyError("number can not be negative")
          else:
              print("positive number",a)
```

Enter a positive number-1

```
-----
MyError                                Traceback (most recent call last)
Cell In[29], line 8
      6 a = int(input("Enter a positive number"))
      7 if(a<0):
----> 8     raise MyError("number can not be negative")
      9 else:
     10     print("positive number",a)

MyError: number can not be negative
```

02) WAP to raise your custom Exception.

```
In [37]: class MyError(Exception):
          def __init__(self):
              self.message = "number cannot be negative"

          a = int(input("Enter a odd number"))
          try:
              if(a%2 == 0):
                  raise MyError
              else:
                  print("positive number",a)
          except MyError as e:
              print(e.message)
```

Enter a odd number4
number cannot be negative

In []: