

Task 9: Implement exceptions and Exceptional handling in python.

(a). student Marks Validation.

Aim:- To write a python program that validates student marks and handles exceptions if the marks are invalid.

Algorithm:-

1. Start the program.
2. Accept marks input from the user.
3. Convert the input into an integer.
4. Check if marks are less than 0 or greater than 100.
5. If so, raise an exception with an appropriate message.
6. Use try-except to catch and display the error message.
7. In valid, display the marks entered.
8. End the program.

Program

student marks validation.

try:

marks = int(input("Enter student marks: "))

if marks < 0 or marks > 100:

raise ValueError("Marks must be between 0 and 100.")

print("Valid marks:", marks)

except ValueError as e:

print("Error:", e).

Input :

Enter student marks : 120.

Output :-

Error: marks. must be. between 0 and 100.

Results: The program successfully validates student marks and displays an error message for invalid input.

(b). Division calculator with exception Handling.

Aims- To write a python program that perform division of two numbers and handles exceptions like division by zero and invalid input.

Algorithm

1. Start the program
2. Use a try block to take two numbers as input.
3. Convert the inputs into float or int.
4. Perform division and display the result.
5. Use except to handle:
 - ZeroDivisionError if denominator is zero.
 - ValueError if the input is not a valid number.

Program

```
# Division. calculator with exception Handling
```

try:

```
num1 = float(input("Enter numerator:"))
```

~~```
num2 = float(input("Enter denominator:"))
```~~~~```
result = num1 / num2.
```~~~~```
print("Result: ", result)
```~~~~```
except ZeroDivisionError:
```~~~~```
print("Error: Division by zero is not allowed.")
```~~~~```
except ValueError:
```~~~~```
print("Error: Invalid input! please. enter numeric values only. ")
```~~

Input 1:

Enter numerator: 10

Enter denominator: 0

Output 1:

Error: Division by zero is not allowed.

Algorithm to convert a decimal fraction to a vulgar fraction  
Step 1: Find the number of digits after the decimal point.  
Step 2: If the number of digits is 1, then the denominator will be 10.  
Step 3: If the number of digits is 2, then the denominator will be 100.  
Step 4: If the number of digits is 3, then the denominator will be 1000.  
Step 5: If the number of digits is 4, then the denominator will be 10000.  
Step 6: If the number of digits is 5, then the denominator will be 100000.  
Step 7: If the number of digits is 6, then the denominator will be 1000000.  
Step 8: If the number of digits is 7, then the denominator will be 10000000.  
Step 9: If the number of digits is 8, then the denominator will be 100000000.  
Step 10: If the number of digits is 9, then the denominator will be 1000000000.  
Step 11: If the number of digits is 10, then the denominator will be 10000000000.

Programs written in this notebook are given as

Input 1: (Enter a decimal fraction)  $\frac{1}{10}$  = 0.1  
Input 2: (Enter a decimal fraction)  $\frac{1}{100}$  = 0.01  
Enter numerator: ten. (Enter a value > than 10)  
Enter denominator: 2. (Enter a value < than 2)  
Output 1: for 0.10000000000000002 well-formed fraction  
Error: Invalid input! please enter numeric values only.

|                         |          |
|-------------------------|----------|
| PERFORMANCE (5)         | 5        |
| RESULT AND ANALYSIS (5) | 5        |
| VIVA VOCE (3)           | 5        |
| REC'D BY (1)            | 1        |
| TOTAL (20)              | 15       |
| SIGN WITH DATE          | 14/10/25 |

Result:

The program correctly performs division and handles both invalid and division-by-zero errors.