

9.	24/09/25	Implement Exception and Exception Handling in python		
10.	24/09/25	Use the Matplotlib Module for Plotting in Python		
11.	08/10/25	Use the Tkinter Module for UI design	15	15/10/25
12.	08/10/25	Simulate gaming concepts Using Pygame		
12.	12/10/25	Pygame		

1. Input marks to a variable 'm' and check if it is between 0 and 100. If not, display an error message and ask for input again. If yes, display the marks.

(1) If marks are between 0 and 100, display the marks.

(2) If marks are not between 0 and 100, display an error message and ask for input again.

Example:

Input: 120

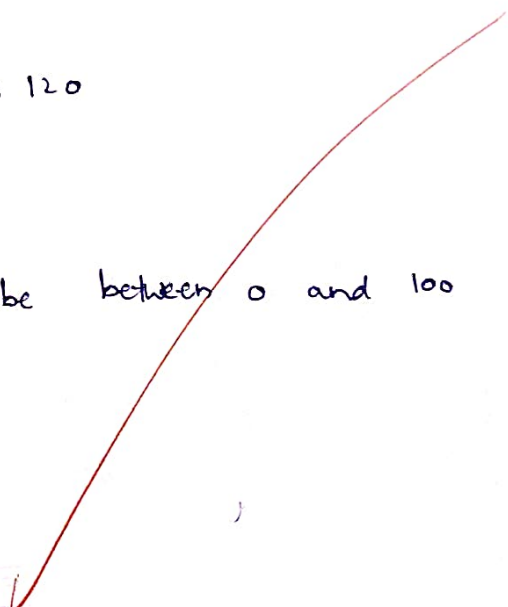
Output: Error! Marks must be between 0 and 100

Input:-

Enter student marks: 120

Output:-

Error! Marks must be between 0 and 100



VEL TECH - CSE	
Sl. No.	
1	PERFORMANCE (2)
2	THEORY AND ANALYSIS (3)
3	VIVA VOCE (5)
4	RECORD (2)
5	TOTAL (120)
DATE	

9/9/25 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. Invalid, 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)

Result:-

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

Input 1:

Enter numerator : 10

Enter denominator : 0

Output 1:

Error: Division by zero is not allowed

Input 2:

Enter numerator : ten

Enter denominator : 2

Output 2:

Error: Invalid input! please enter numeric values only.

24/9/25

(b) Division Calculator With Exception Handling

Aim: To write a python program that performs 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")
```

VEL TECH - CSE	
EX NO.	9
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	20
SIGN WITH DATE	

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

15/10/25