

TASK-9: IMPLEMENT EXCEPTIONS AND EXCEPTIONAL HANDLING IN PYTHON.

22/9/25

Aim:

TO implement Exceptions and Exceptional handling in Python.

PROBLEM 9.1: Program is designed to allow the user to select a grade by specifying an index number. However, you need to ensure that the program handles cases where the user inputs an index that is out of range, i.e., an index that does not exist in the list.

ALGORITHM:

1. Start the program.
2. Initialize a list of grades (e.g., [85, 90, 78, 92, 88]).
3. Prompts the user to enter the index of the grade they wish to view.
4. Attempts to display the grade at specified index.
5. Please enter the valid index.

PROGRAM:

```
# Initialize the list of grades
grades = [85, 90, 78, 92, 88]
```

```
# Display the grades list
```

```
Print ("Grades list:", grades)
```

```
# Prompt the user to enter the index of the grade they want to view
try:
```

```
Index = int (Input ("Enter the index of the grade you want to view:"))
```

```
# Attempt to display the grade at specified index
```

```
Print (f"The grade at index {Index} is: {grades[Index]}")
except IndexError:
```

```
# Handle the case where the index is out of range  
print("Invalid index. Please enter a valid index.")
```

```
except ValueError:
```

```
# Handle the case where the input is not an integer  
print("Invalid input. Please enter a numerical index.")
```

PROBLEM: 9.2 One of the key functionalities is to divide two numbers entered by the user. However, dividing by zero is not allowed & would cause the program to crash if not handled properly.

ALGORITHM:

1. Start the program
2. Prompts the user to enter two numbers: a numerator & a denominator.
3. Attempts to divide the numerator by the denominator.
4. If the denominator is zero, catches the zero division error and displays an error message:
"Error: Division by zero is not allowed."

PROGRAM:

```
# Function to perform division  
def divide_numbers():  
    try:  
        # prompt the user to enter the numerator  
        numerator = float(input("Enter the numerator: "))  
        # prompt the user to enter the denominator  
        denominator = float(input("Enter the denominator: "))  
        # attempt to perform division  
        result = numerator / denominator
```

OUTPUT:

Grades list : [85, 90, 78, 92, 88]
Enter the index of the grade you want
to view : 10
Invalid index. Please enter a valid index.

APPENDIX:

1. How to store the list of grades.
2. How to store the index of the grade.
3. How to store the index of the grade.
4. How to store the index of the grade.

APPENDIX:

How to store the list of grades.

How to store the index of the grade.

How to store the index of the grade.

How to store the index of the grade.

How to store the index of the grade.

How to store the index of the grade.

How to store the index of the grade.

How to store the index of the grade.

How to store the index of the grade.

How to store the index of the grade.

How to store the index of the grade.

OUTPUT: Enter the numerator: 10
Enter the denominator: 0
Error: Division by zero is not allowed

if (denominator == 0) {
 cout << "Error: Division by zero is not allowed" << endl;
 return 0;
}

cout << "Result: " << (float) numerator / denominator << endl;

return 0;

if (denominator == 0) {
 cout << "Error: Division by zero is not allowed" << endl;
 return 0;
}

cout << "Result: " << (float) numerator / denominator << endl;

return 0;

if (denominator == 0) {
 cout << "Error: Division by zero is not allowed" << endl;
 return 0;
}

cout << "Result: " << (float) numerator / denominator << endl;

return 0;

if (denominator == 0) {
 cout << "Error: Division by zero is not allowed" << endl;
 return 0;
}

cout << "Result: " << (float) numerator / denominator << endl;

return 0;

if (denominator == 0) {
 cout << "Error: Division by zero is not allowed" << endl;
 return 0;
}

cout << "Result: " << (float) numerator / denominator << endl;

```

Print ("Result : {result}")
except Zero Division Error:
    # Handle division by zero error
    print ("Error: Division by zero is not allowed.")
except value error:
    # Handle invalid input that is not a number
    Print ("Error: Please enter valid numbers.")
# call the function to execute the division operation
divide - numbers ()

```

| VELTECH | |
|-------------------------|--|
| EX No. | |
| PERFORMANCE (5) | |
| RESULT AND ANALYSIS (5) | |
| VIVA VOCE (5) | |
| PROJECT (5) | |
| TOTAL (20) | |
| DATE | |

RESULT: Thus the program for implement Exceptions and Exceptional handling is executed & verified Successfully.