

Task 9: Implement Exceptions & Exceptional handling in Python

17/9/25

Aim To implement Exceptions & Exceptional handling in Python.

Algorithm:

1. Start the program
2. Initializes a list of grades [85, 90, 78, 92, 88]
3. Prompt the user to enter index of grade wished to view.
4. Attempts to display the grade at the specified index.
5. If the index is out of range, catches the IndexError and prints an error message.

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 grade wished to view:
# 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]}")

try:
    index = int(input("Enter the index of the grade you want to view:"))
    print(f"The grade at index {index} is {grades[index]}")
except IndexError:
    # Handle the case where the index is out of range
    print("Invalid range. Please enter a valid index.")

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

Output

Grades lists: [85, 90, 78, 92, 88]

Enter the index of the grades you want view: 10

Invalid index. Please enter a valid index

Task 92, To develop a python calculator program

Aim: To develop a python calculator program that perform basic arithmetic operation

Algorithm:

1. Start the program
2. Prompts the user to enter two numbers a numerator denominator
- 3 Attempts to divide the numerator by denominator
- 4 If the denominator is zero, catches the ZeroDivisionError and displays an error message:

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
        print(f"Result: {result}")
    except ZeroDivisionError:
        print("Error: Division by zero is not allowed")
    except ValueError:
        # Handle invalid input that is not number
        print("Error: Please enter valid numbers!")
# Call the function to execute the division operation
divide_numbers()
```

Output:

Enter the numerator: 10

Enter the denominator: 0

ERROR!

Error: Division by zero is not allowed

Output

Enter a number : 15

Exception occurred: Invalid Age

Task 9.3

Aim: To build a Python application to determine.

Algorithm:

1. Define the custom exception.
2. Prompt the user of input.
3. Check if the age is below 18.
4. Raise an exception if the condition is met.
5. Handle the exception with

Program:

```
#define Python user-defined exceptions
class InvalidAgeException(Exception):
    "Raised when the input value is less than 18"
    pass
# You need to guess this number
number = 18

try:
    input_num = int(input("Enter a number: "))
    if input_num < number:
        raise InvalidAgeException
    else:
        print('Eligible to Vote')
except InvalidAgeException:
    print("Exception occurred")
```

PERFORMANCE (5)	
EX NO.	9
PERFORMANCE (5)	5
BEST PRACTICE AND ANALYSIS (5)	5
VIVA VOCE (3)	3
REPORT (4)	5
TOTAL (15)	15
SIGN WITH DATE	

Result: Thus the program for Implement Exceptions and Exception handling is executed and verified successfully