Date: 24/9/25

Task 9: Implement Expectations and Exceptional handling in Python.

Aim: To implement Exceptions and Exceptional handling in Python.

9.1 *. You are developing a python program that processes a list of students' grades. The 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. Initializes 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 the specified index.
5. If the index is out of range, catches the IndexError and prints an error message, "Invalid index. Please enter a 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 the 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.")
```

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.

9.2 You are developing a python calculator program that performs basic arithmetic operations. one of the key functionalities is to divide two numbers entered by the user. However, dividing by zero is not allowed and could cause the program to crash if not handle properly.

Algorithm:

1. Start the program
2. Prompts the user to enter two numbers: a numerator and a denominator
3. Attempts to divide the numerator by the denominator.
4. If the denominator is zero, catches the ZeroDivision 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
    Print(f"Result: {result}")
  except ZeroDivisionError:
    # Handle division by zero error
    Print("Error: Division by zero is not allowed.")
  except ValueError:
    # Handle invalid input that is not a 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.

9.3: You are building a python application to determine is a person is eligible to vote based on their age. According to the rules, only individuals who are 18 years or older are allowed to vote. To enforce this rule, you decide to create a custom exception called InvalidAgeException, which will be raised whenever an age below 18 is entered.

Algorithm:

1. Define the custom exception.
2. Prompt the user for input.
3. check if the age is below 18.
4. Raise an exception if the condition is met.
5. handle the exception with a custom error message.

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 occured: Invalid age")
```

VEL TECH	
EX NO.	9
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	5

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

Output

Enter a number: 15

Exception occured: Invalid age