22/9/25 Task-(9). Implement Exceptions and Exceptional handling In Python.

Aim: To implement Exceptions and exceptional in handling python.

Algorithm :-

1. start the program.
2. Initializes list of grade ([150, 18, 90, 92, 88])
3. prompts the user enter the index of grade they wish to view.
4. Attempts to display out of range, catches 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 index of grade they want to view
try:
  index = int(input("Enter the index of grade you want to view"))
# Attempt to display the grade at specified index
  print (f"The grade at index {index} is: {grade[index]}")
except IndexError
  # Handle the case where the index is out of range.
  print ("Invalid index. Please enter a valid index.")
except value Error:
  # Handle the case where 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 grade you want to view: 0.

Invalid index please enter a valid index.

output:

enter the numerator:10
enter the denominator:0
Error!
Error: Division by zero is not allowed.

:- 9.2

You are developing a python calculator program perform basic arithmetic operations.

Algorithm:

1. start the program.
2. prompts the user to enter two number : a numerator and a denominator.
3. Attempts to divide the numerator by denominator.
4. If the denominator is zero, catches the zero Division Error & displays an error message:

Program:

```
# Function to perform division:
def divide_numbers():
try:
# prompt the user enter the numerator.
numerator = float(input("Enter the numerator:"))
# Attempt to perform division.
result = numerator / denominator.
print(f"Result:{result}")
Except zero bivision 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().
```

output:

Enter the number:15

Exception occured : Invalid Agl.

(Q.3) Algorithm

Algorithm:

1. Define the custom exception:
2. Prompt the user for input.
3. Check if 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 InvalidAge Exception(Exception):
    "Raised when the input value is less than 18".
    pass.

# you need the guess this number.
number = 18.
try:
  input_num = int(input("Enter a number: "))
  if input_num < number.
      raise invalid exception.
  else:
    print("Eligible to vote").
  except InvalidAge Exception:
    print("exception occured : Invalid Age").
```

Result: thus, the program for implement exceptions and exceptional handling is executed and verified successfully.

EX.NO.	
PERFORMANCE (5)	
RESULT AND ANALYSIS (5)	
VIVA VOCE (5)	
RECORD (5)	
TOTAL (20)	
SIGN WITH DATE	