10/9/25

Task 9: Implement Exceptions and exceptiona handling

Aim: To implement exceptions and exceptional handling in python.

Algorithm:

```
1. Start the Program
2. Initializes a list of grades (eg:[85,90,70,92,88])
3. Prompts the users 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 index Error and prints an error message,
   "Invalid index Please enter a valid index!"
```

Program:

```
# initalize 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 is: {grades[index]}")
  except Index Error:
     # 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 the inputs is not an integer
  Printf("Invalid input, Please enter a numerical index")
```

Output:

```
Grades list: (85, 90, 78, 92, 88)
Enter the Inden of the grade you want to vi
Invalid index Please enter a valid index.
```

q.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 :.

Algorithm:-

1. start.
2. Prompts the user to Enter two numbers; a numerator and d denominator.
3. Attempts to divide the numerator by the denominator.
4. If the denominator is zero, catches the zero division Error and display an error message. "Error": Division by zero is not allowed".

Program:-

```
def divide_numbers( ):

    try:
     numerator = float (input ("Enter the numerator:"))
     denominator = float (input ("Enter the denominator:"))
     result = numerator / denominator.
      Print(f"Result: {Result}")
   except Zero Division error:
      Print ("Error: please Enter valid numbers:")
 divide_numbers( )
```

q.3 You are building a Python application to determine if a person is figible to role bases on their age. According to the rules, only individuals who are 18 years or older are allowed to vote.

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.

Output:-

```
Enter the number : 10
enter the denominator : 0
  ERROR !
ERROR: Division by zero is not allowed
```

✓

Program:

```
class Invalid Age exception (exception):
    " Raised when the input value is less than 18"
    pass
number = 18
try:
    input_num = int(input ("Enter a number:"))
    if input_num < number
    raise Invalid Age Exception
    else:
        print("Eligible to vote")
except Invalid Age exception:
    print("Exception occured : Invalid Age")
```

Result: Thus, the Program for Implement exceptions and exceptional handling is ~~ex~~ executed and verified successfully.

VEL TECH	
No.	9
RFORMANCE (5)	5
LT AND ANALYSIS (5)	5
VOCE (5)	5
RD (5)	5
TAL (20)	20
H DATE	8 25/10

Output:

```
Enter a number : 15
exception occured :Invalid Age.
```