

## Task-9 :- Implement Exceptions and Exceptional handling

Aim :- To implement Exceptions and Exceptional handling in python.

### Algorithm:-

1. Start the program
2. Initializes a list of grades (eg; [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 Index Error 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.  
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").
```

### Problem - 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.

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:

X

AEL TECH	
1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	38
39	40
41	42
43	44
45	46
47	48
49	50
51	52
53	54
55	56
57	58
59	60
61	62
63	64
65	66
67	68
69	70
71	72
73	74
75	76
77	78
79	80
81	82
83	84
85	86
87	88
89	90
91	92
93	94
95	96
97	98
99	100

### Algorithm :-

1. Starts 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 zero Division Error and displays an error message : "Error : Division by zero is not allowed"

### Program :-

# Function to perform division.

```
def divide_numbers():
```

    # By :

```
        # 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()
```

Problem :- q.3 :- you need one building a python application to determine

if 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.

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.

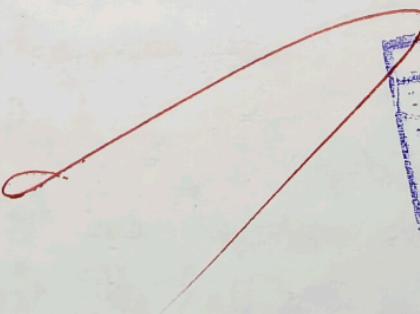
### Algorithm:-

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 occurred: Invalid Age")
```

Result:- Thus, the program for implementing Exceptions and Exceptional handling is executed and verified successfully.



VELTECH	
PERFORMANCE (5)	9
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	15
TOTAL (20)	17.5
SIGN WITH DATE	