Task9:- Implement Exceptions and exceptional handling in Python

Aim:- To implement Exceptions and exceptional handling in Python.

Problem 9.1: you are developing a python program that process a list of student's grades the program is designed to allow the user to select a grade by specifing an index number However you need to ensure that the program handles cases where the user input an index that is out of range an index that does not exist the list.

Algorithm:-

1) Start the program
2) initialize the list of grades [85, 90, 78, 92, 88]
3) Prompts the user to enter the index of the grade they wish to view
4) Attempt to display the grade at the specified index
5) if the index is out of range, catches the index Error and print an error massage "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:
```

output:-

```
Grades lists: [85, 90, 78, 92, 88]
Enter the index of the grade you want to
view: 10
Invalid index please enter a valid index
```

index = int [input ["Enter the index of the grade you want to view:"]]

Attempt to display the grade at the specified index

printf("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 value error'

Handle the case where the input is not an integer

~~print~~print ["Invalid input ~~plese~~ please enter a numerical index"]

Problem 9.2 You are developing a python calculator program that process a list of student's grades the program is designed to allow the user to select a groude by specifing an ~~idex~~ index numbey However, you need to ensure that the program handles coses where the user input an index that is out of range an index that does not exist the list.

Algorithm:-

1) Start the program
2) Prompts the users to enter two num a numerator and a deromінator
3) Attempts to divide the numerator by the deromінator
4) if the denoinator is zero catches the zero Division error and display an error message

output:-

```
Enter the numerator :10
Enter the denominator:0
ERROR!
Error: Division by zero is not allowed.
```

"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 numereator"]]
  # Pramp the user to enter the denominator
    denominator = float [input ["Enter the deminator"]]
  # Attempt to perform division
      result = numerator / denominator
        Print [f"Result: {result}"]
        except zero Division error:
    # Handle division by zero error
        Print ["Error: Division by zero is not allowed"]
    # call the function to execute the division
        operation divide - numbers()
```

Problem 9.3:- You are building a python application to determine if a person is eligible to vote based on their age or older are allowed to vote ~~the~~ TO enfore this nues you decide to create a custom exception called invalid age exception, which will be raised whenever an age below 18 is entered.

Algorithm:-

1) Define the cus form exception
2) Prompt the user for input
3) check if the age is below 18

output:-

Enter a number: 15

Exception occurred : invalid Age.

5) Handle the exception with a wstom error mesge

Program:-

```
# define Python user-define exceptions
class Invalid Age exception 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:
raised invalid Age Exception
else;
Print("Eligible to vote");
exceptinvalid Exception:
Print ["Exception occurred: invalidage"]
```

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

Result:- Thus the program implement exceptions and exceptional handling is executed and verified success fully.