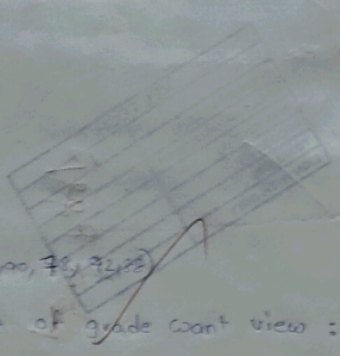
Program:-

```
grades = [85, 90, 78, 92, 88]
print("grades list:", grades)
grade they want to view
try:
    index = int(input("Enter the index of grade you want to view"))
    print(f"The grade at index {index} {grade index}")
    print("Invalid index please enter a index")
except value error:
    Print ("Invalid input please enter a numerical index")
```



output:-

Grades list [85, 90, 78, 92, 88]

Enter the index of grade want view : 10

Invalid index please enter a valid index.

Date: 22/9/25

TASK-9 Implement exceptions and exceptional handling System:-

Aim :- To implement exceptions and exceptional handling in python.

Problem:-① You are developing a python program that process a list of students grades. The program is designed to allow the user a select a grade by specifying an index no. However an index that does not exist in list.

Algorithm:-

① Start the Program.

② Initializes a list grades (eg:[85, 90, 78, 92, 88])

③ Prompts the user to enter the index of grade wish to view

④ Attempts to display grade at 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:-

```
def divide_numbers():
    try:
        numerator = float(input("Enter numerator:"))
        denominator = float(input("Enter denominator:"))
        result = numerator / denominator
        print(f"Result : {result}")
    except ZeroDivisionError:
        print("error : Division by zero is not allowed")
    except ValueError:
        print("Error: Please enter valid no's")
divide_numbers()
```

Outputs:-

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

(9.2) You are developing a python calculator prog:- -am that performs basic arithmetic operations. one of the key functionalities allowed and could cause the program to crash if not handle properties.

Algorithm

1. Start the program.
2. Prompts the use to enter two number a numer- -ator and a denominator.
3. Attempts to divide the numerator by the denomi- -nator.
4. If the denominator is zero catches the messa- -ge :"Error": division by zero is not allowed.

Program:-

```
class InvalidAgeException(Exception):
    "Raised when the input value is less 18"
    pass
"you need to guess this number"
number=18
try:
input("name number:")
raise InvalidAgeException
else:
Print("Eligible to vote")
except InvalidAgeException:
print("Exception occured: Invalid Age")
```

Outputs:-

```
Enter a number: 15
Exception occured: Invalid Age
```

9.3 You are 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. To enforce this rule whenever

Algorithm:- an age below 18 is entered.

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 custom error message.

VELTECH	
EX No	9
PERFORMANCE (5)	05
RESULT AND ANALYSIS (5)	05
VIVA VOCE (5)	
RECORD (5)	
TOTAL (20)	10
SIGN WITH DATE	

Result:- Thus, the program for implement exception handling is executed and verified successfully.