

Task 9: Implement Exceptions and Exceptional handling in Python

Aim: To implement Exceptions and exceptional handling in Python

Algorithm:

1. start the Program
2. initializes a list of grades
3. prompts the user to enter the index of grade
4. attempts to display the grade at the specific index

Program

```
# initialize the list of grades
```

```
grades = [85, 90, 78, 42, 88]
```

```
print("grades list:", grades)
```

```
index = int(input("Enter the index of grade you want to view:"))
```

```
print(f"the grade at index {index} is: {grades[index]}")
```

```
except IndexError
```

```
print("Invalid index please enter a valid index")
```

```
except ValueError
```

```
print("invalid input please enter a numerical index")
```

(positive) long

(double) long

(std::vector) long

Output:

Grade list: [85, 90, 78, 92, 88]

Enter the index of the grade you want to view: 10

Invalid index. Please enter a valid index

ANSWER

Q	EX 10.
2	CHALLENGE (6)
2	CHARACTERISTICS (6)
2	(f) CONCLUSION (6)

Q.2 You are developing a Python calculated program that performs basic arithmetic operations

of type

Algorithm:

1. Start the Program
2. Prompts the user to enter two numbers
3. Attempts to divide the numerator by denominator
4. If the denominator is zero, catches the zero division error

Program

```
Def Divide - numbers():
```

try

```
numerator = float(input("Enter the numerator:"))
```

```
denominator = float(input("Enter the denominator:"))
```

```
result = numerator/denominator
```

~~Print ("Result: " + result)~~

~~except ZeroDivisionError:~~

~~Print ("Error: Division by zero is not allowed")~~

~~except ValueError:~~

~~Print ("Error: Please enter valid number")~~

Output in print board

Output:

Enter the numerator: 10

Enter the denominator: 0

Error!

~~Error: Division by zero is not allowed~~

Q3: 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 older are allowed to vote. To enforce this rule you decided to create a custom exception called Invalid Age exception.

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

Program:

```
class InvalidAgeException(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 InvalidAgeException
```

```
    else:
```

```
        print("eligible")
```

```
except InvalidAgeException:
```

```
    print("exception occurred : invalid age")
```

VEL TECH	
EX NO.	87
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
AV A VOCE (5)	5
RECORD (5)	5
GRADUATION (%)	100
GRADUATION DATE	15/07/2023

Result: thus the program for implementing exceptions and Exceptional handling is executed and Verified successfully