

Adhish Bahl

2347203

1MCA B

Python Lab

Lab 5

Q1) Write a program to handle the exception of ZeroDivisionError.

```
def add(a, b):
    sum = a + b
    print(a, "+", b, "=", sum)

def subtract(a, b):
    difference = a - b
    print(a, "-", b, "=", difference)

def multiply(a, b):
    product = a * b
    print(a, "x", b, "=", product)

def divide(a, b):
    try:
        division = a / b
        print(a, "/", b, "=", division)
    except ZeroDivisionError as e:
        print("Devision by 0 is not allowed")
    finally:
        print("Thank you")

choice = 0
while choice != 5:
    print("\nMENU")
    print("1. Sum of two Numbers")
    print("2. Difference between two Numbers")
    print("3. Product of two Numbers")
    print("4. Division of two Numbers")
    print("5. Exit")
    choice = int(input("\nEnter the Choice: "))

    a = int( input("First Number: "))
    b = int( input("Second Number: "))

    if choice == 1:
        print( "\nADDITION")
        add(a, b)
    elif choice == 2:
```

```
        print( "\nSUBTRACTION")
        subtract(a, b)
    elif choice == 3:
        print( "\nMULTIPLICATION")
        multiply(a, b)
    elif choice == 4:
        print( "\nDIVISION")
        divide(a, b)
    elif choice == 5:
        break
    else:
        print( "Please Provide a valid Input!")
```

MENU

1. Sum of two Numbers
2. Difference between two Numbers
3. Product of two Numbers
4. Division of two Numbers
5. Exit

ADDITION

10 + 0 = 10

MENU

1. Sum of two Numbers
2. Difference between two Numbers
3. Product of two Numbers
4. Division of two Numbers
5. Exit

DIVISION

Devision by 0 is not allowed
Thank you

MENU

1. Sum of two Numbers
2. Difference between two Numbers
3. Product of two Numbers
4. Division of two Numbers
5. Exit

MULTIPLICATION

10 x 3 = 30

MENU

1. Sum of two Numbers
2. Difference between two Numbers
3. Product of two Numbers
4. Division of two Numbers

5. Exit

SUBTRACTION

10 - 20 = -10

MENU

1. Sum of two Numbers
2. Difference between two Numbers
3. Product of two Numbers
4. Division of two Numbers
5. Exit

Q2) Write a program to handle the exception of IndexError.

```
def getElement(data_list, index):
    try:
        value = data_list[index]
        return value
    except IndexError:
        return "Index Error"
    finally:
        print("\nValue Sent!")

myList = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
while True:
    try:
        index = int(input("Enter an position to retrieve data (-1 to exit): "))

        if index == -1:
            print("Exiting the program.")
            break
        element = getElement(myList, index-1)
        if element == "Index Error":
            print("Index out of range. Please enter a valid index.")
        else:
            print("Value at index", index, "is:", element)

    except ValueError:
        print("Invalid input. Please enter a valid index.")
```

Value Sent!

Value at index 2 is: 20

Value Sent!

Value at index 10 is: 100

Value Sent!

Value at index 5 is: 50

Value Sent!

Index out of range. Please enter a valid index.

Value Sent!

Index out of range. Please enter a valid index.

Exiting the program.

Lab 6

Q1) Write a program using the Regular Exception and create a function that accepts a string and searches it for a valid phone number.

Return the phone number if found.

A valid phone number may be one of the following:

(xxx)-xxx-xxxx

xxx-xxx-xxxx

```
import re

def searchNumber(text):
    phonePattern = r'\b(\d{3})-\d{3}-\d{4}|\d{3}-\d{3}-\d{4})\b'
    match = re.search(phonePattern, text)
    if match:
        return match.group(0)
    else:
        return None
```

```
inputText = input("Enter a string: ")
phoneNumber = searchNumber(inputText)
```

```
if phoneNumber:
    print("Your String: ", inputText, "\nValid phone number found:",
          phoneNumber)
else:
    print("Your String: ", inputText, "\nNo valid phone number found
in the string.")
```

```
Your String: Hello, this is Adhish Bahl. My number is (946-284-9725).
I like Python.
Valid phone number found: 946-284-9725
```

Q2. Write a function that employs regular expressions to ensure the password given to the function is strong.

A strong password is defined as follows:

- at least eight characters long
- contains one uppercase character
- contains one lowercase character
- has at least one digit
- has at least one special character

[For instance: [Christ@123](#)]

```
import re

def checkPassword(password):
    passwordPattern = r"^(?=.*[A-Z])(?=.*[a-z])(?=.*\d)(?=.*[@#$%^&+=!]).{8,}$"

    if re.match(passwordPattern, password):
        return True
    else:
        return False

password = input("Enter a password to check if it's strong: ")
if checkPassword(password):
    print("Your password \'", password, "\' is strong.")
else:
    print("Your password \'", password, "\' is not strong. It should have at least 8 characters, one uppercase letter, one lowercase letter, one digit, and one special character.")
```

```
Your password ' Adhishbahl. ' is not strong. It should have at least 8 characters, one uppercase letter, one lowercase letter, one digit, and one special character.
```