

Lab Exercise 5

1. Write a program to handle the exception of ZeroDivisionError.

```
In [ ]: def main():
    try:
        num1 = float(input("Enter a number: "))
        num2 = float(input("Enter another number: "))
        if num2 == 0:
            raise ZeroDivisionError("Cannot be divided by zero")
        result = num1 / num2
        print("Result:", result)
    except ZeroDivisionError as error:
        print("Error:", error)
    except ValueError:
        print("Invalid input. Please enter valid numbers.")
    except Exception as e:
        print("An error occurred:", e)

main()
```

Error: Cannot be divided by zero

2. Write a program to handle the exception of IndexError.

```
In [ ]: def execute():
    try:
        lst = [1, 2, 3]
        idx = int(input("Enter an index: "))

        val = lst[idx]
        print("Value at index", idx, "is:", val)

    except IndexError:
        print("Index out of range.")
    except ValueError:
        print("Invalid input. Please enter a valid index.")
    finally:
        print("Execution completed.")

execute()
```

Value at index 2 is: 3
Execution completed.

Lab Exercise 6

1. 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

```
In [ ]: import re

def find_phone_number(number):
    pattern = r'\b\d{3}-\d{3}-\d{4}\b'
    match = re.search(pattern, number)

    if match:
        return match.group()
    else:
        return None

def main():
    input_text = input("Enter a text to search for a valid phone number: ")
    phone_number = find_phone_number(input_text)

    if phone_number:
        print("Valid phone number found:", phone_number)
    else:
        print("No valid phone number found in the input text.")

main()
```

Valid phone number found: 790-515-5089

2. 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]

```
In [ ]: import re

def is_strong_password(password):
    pattern = r"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}"
    return re.match(pattern, password) is not None

def main():
    password = input("Enter a password: ")
    print(password)
    if is_strong_password(password):
        print("The password is strong.")
    else:
        print("The password is not strong.")

main()
```

Christ@123

The password is strong.