ARTIFICIAL INTELLIGENCE LAB

BSCYS-3rd Semester

Fall 2025

Lab Report # 3

Submitted: To: Sir Mubashir Iqbal

Submited By: M.Umer

Reg No: (24-CyS-024)

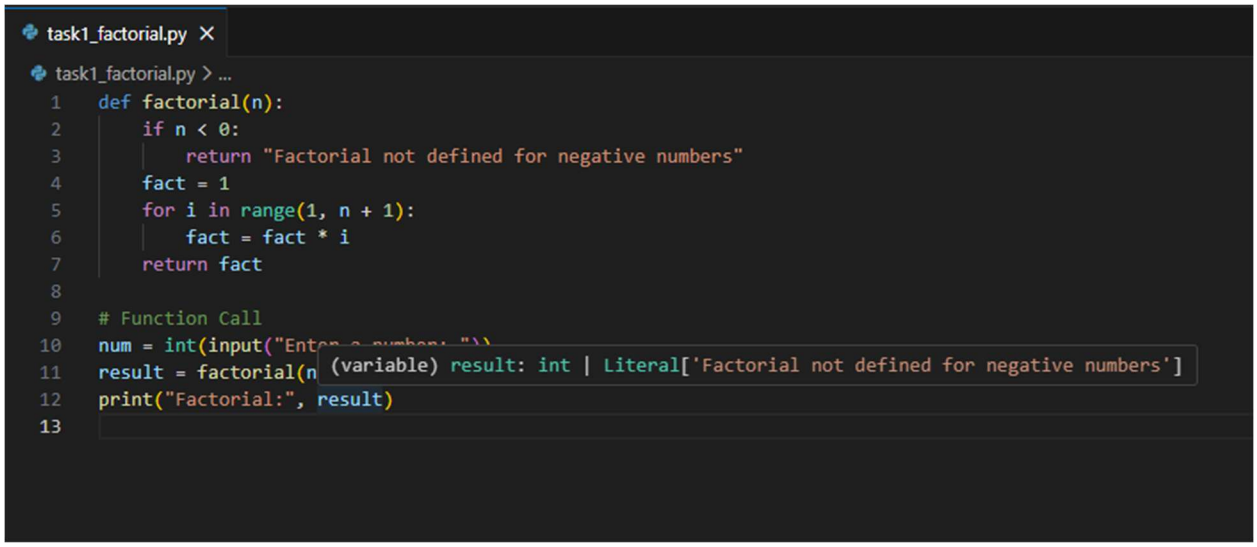
Department: Cyber Security (A)

DEPARTMENT OF COMPUTER SCIENCE HITEC UNIVERSITY TAXILA

BS CYBER SECURITY PROGRAM

Task 1:

Code:



Output:

Task 2:

Code:

task1_factorial.py task2_unique_list.py

task2_unique_list.py > ...

```python
def unique_list(lst):
    unique = []
    for item in lst:
        if item not in unique:
            unique.append(item)
    return unique

# Sample List
sample_list = [1,2,3,3,4,4,5,3,7,5]
print("Sample List:", sample_list)
print("Unique List:", unique_list(sample_list))

```

Output:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS C:\Users\LAB 06\Desktop\AICT> & "C:/Users/LAB 06/AppData/Local/Programs/Python/Python312/python.exe" "c:/Users/LAB 06/Desktop/AICT/task2_unique_list.py"
Sample List: [1, 2, 3, 3, 4, 4, 5, 3, 7, 5]
Unique List: [1, 2, 3, 4, 5, 7]
PS C:\Users\LAB 06\Desktop\AICT>
```

Task 3:

Code:

task1_factorial.py | task2_unique_list.py | task3_even_numbers.py

task3_even_numbers.py > ...

```python
def even_numbers(n):
    evens = []
    for i in range(1, n + 1):
        evens.append(2 * i)
    print("Even Numbers:", evens)
    return evens

# Function Call
n = int(input("Enter value of n: "))
even_list = even_numbers(n)

# Sum and Product outside the function
total_sum = sum(even_list)
product = 1
for num in even_list:
    product *= num

print("Sum of even numbers:", total_sum)
print("Product of even numbers:", product)

```

Output:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS C:\Users\LAB 06\Desktop\AICT> & "C:/Users/LAB 06/AppData/Local/Programs/Python/Python312/python.exe" "c:/Users/LAB 06/Desktop/AICT/task3_even_numbers.py"
Enter value of n: 2
Even Numbers: [2, 4]
Sum of even numbers: 6
Product of even numbers: 8
PS C:\Users\LAB 06\Desktop\AICT>
```

Task 4:

Code:

task1_factorial.py | task2_unique_list.py | task3_even_numbers.py | task4_string_matching.py

task4_string_matching.py > ...

```python
def match_strings(str1, str2):
    if str1 == str2:
        print("Strings Match")
    else:
        print("Strings Do Not Match")

# Input
string1 = input("Enter first string: ")
string2 = input("Enter second string: ")

# Function Call
match_strings(string1, string2)

```

Output:

Task 5:

Code:

task1_factorial.py | task2_unique_list.py | task3_even_numbers.py | task4_string_matching.py | task5_user_login.py

task5_user_login.py > ...

```python
def login(username, password):
    correct_username = "admin"
    correct_password = "1234"

    if username == correct_username and password == correct_password:
        print("Login Successful")
    else:
        print("Invalid Username or Password")

# User Input
user = input("Enter username: ")
pwd = input("Enter password: ")

# Function Call
login(user, pwd)

```

Output:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS C:\Users\LAB 06\Desktop\AICT> & "C:/Users/LAB 06/AppData/Local/Programs/Python/Python312/python.exe" "c:/Users/LAB 06/Desktop/AICT/task5_user_login.py"
Enter username: admin
Enter password: 1234
Login Successful
PS C:\Users\LAB 06\Desktop\AICT>
```

Overall conclusion:

In this lab, we learned how to create and use **Python functions** to make programs modular, reusable, and organized. We implemented tasks including calculating the **factorial of a number**, extracting **unique elements from a list**, generating the first **n even numbers** and calculating their sum and product, **matching two strings**, and creating a simple **user login system**. Each task demonstrated different aspects of functions such as **required arguments, keyword arguments, default arguments, variable-length arguments, and the return statement**. Through these exercises, we practiced **taking user input, processing data inside**

functions, and returning or displaying results, which helped us understand how functions improve program structure and make Python code efficient and easy to maintain.