COMSATS University Islamabad

Attock Campus

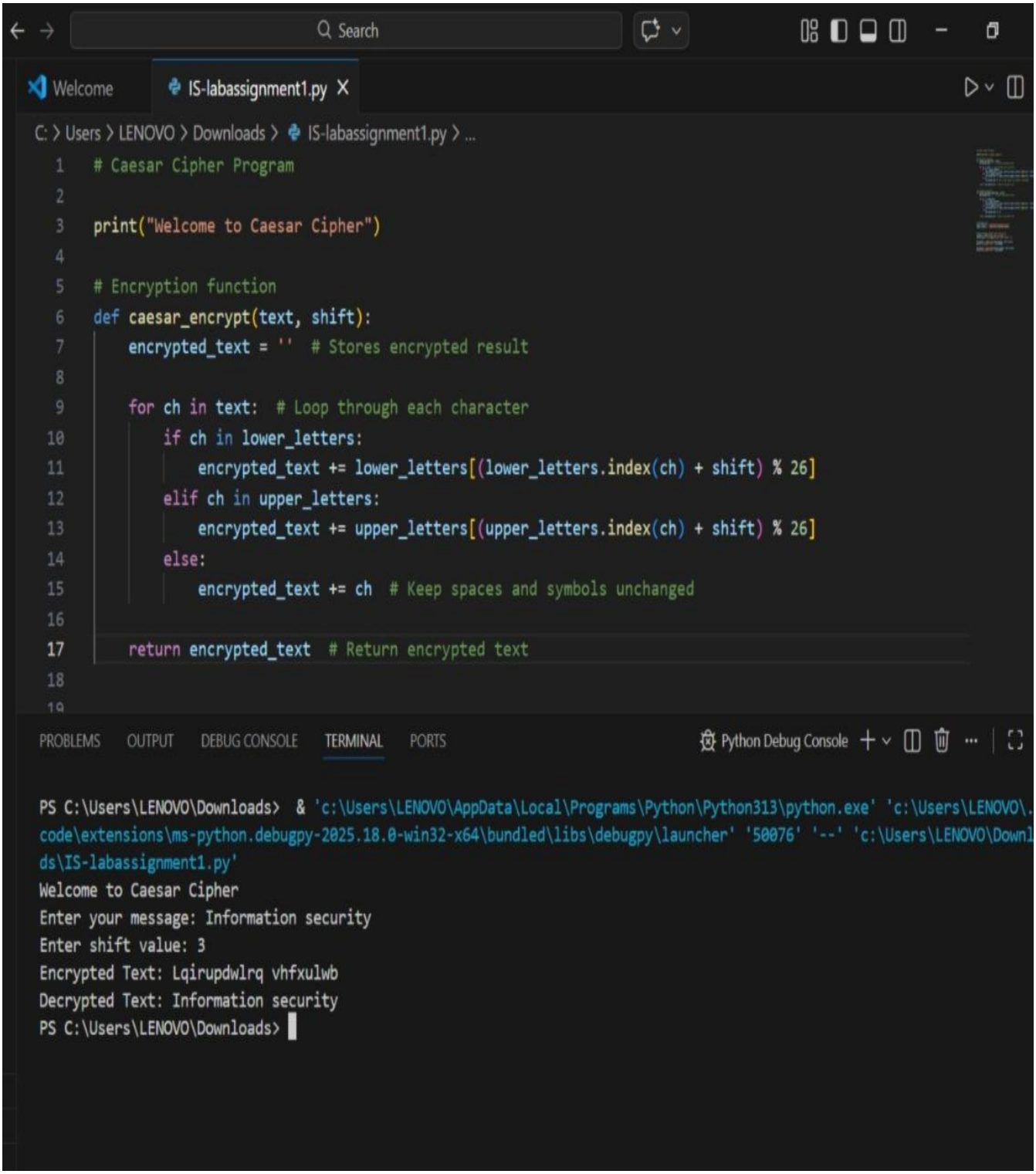
Department Of Computer Science

INFORMATION SECURITY ASSIGNMENT 1

Student Details

Registration No.	Name
FA24-BSE-032	MUHAMMAD Umair

CODE:


Search
Welcome
IS-labassignment1.py
C: > Users > LENOVO > Downloads > IS-labassignment1.py > ...
# Caesar Cipher Program
print("Welcome to Caesar Cipher")
# Encryption function
def caesar_encrypt(text, shift):
encrypted_text = '' # Stores encrypted result
for ch in text: # Loop through each character
if ch in lower_letters:
encrypted_text += lower_letters[(lower_letters.index(ch) + shift) % 26]
elif ch in upper_letters:
encrypted_text += upper_letters[(upper_letters.index(ch) + shift) % 26]
else:
encrypted_text += ch # Keep spaces and symbols unchanged
return encrypted_text # Return encrypted text
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python Debug Console
PS C:\Users\LENOVO\Downloads> & 'c:\Users\LENOVO\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\LENOVO\.
code\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '50076' '--' 'c:\Users\LENOVO\Down
ds\IS-labassignment1.py'
Welcome to Caesar Cipher
Enter your message: Information security
Enter shift value: 3
Encrypted Text: Lqirupdwlrq vhfxulwb
Decrypted Text: Information security
PS C:\Users\LENOVO\Downloads>

Line-by-Line Explanation

1. print("Welcome to Caesar Cipher")
    - Displays a welcome message to the user.
2. # Encryption function
    - Comment explaining that the following code will handle encryption.
3. def caesar_encrypt(text, shift):
    - Defines the function caesar_encrypt with two parameters:
        - text → the message to encrypt
        - shift → number of positions to shift letters
4. encrypted_text = "" # Stores encrypted result
    - Creates an empty string to hold the encrypted message.
5. for ch in text: # Loop through each character
    - Loops through every character in the message.
6. if ch in lower_letters:
    - Checks if the current character is a lowercase letter.
7. encrypted_text += lower_letters[(lower_letters.index(ch) + shift) % 26]
    - Finds the index of the lowercase letter.
    - Adds the shift value.
    - % 26 wraps around if the shift goes past 'z'.
    - Adds the resulting letter to encrypted_text.
8. elif ch in upper_letters:
    - Checks if the character is an uppercase letter.
9. encrypted_text += upper_letters[(upper_letters.index(ch) + shift) % 26]
    - Finds the index of the uppercase letter, adds shift,

wraps around using % 26, and appends it.

10. else:
    - Handles all other characters (spaces, symbols, numbers).
11. encrypted_text += ch # Keep spaces and symbols unchanged
    - Adds the character as-is to the encrypted message.
12. return encrypted_text # Return encrypted text
    - Returns the fully encrypted message after the loop ends.

Security Analysis of Caesar Cipher

The Caesar Cipher is a very simple method of hiding messages by shifting letters in the alphabet. While it is easy to understand and implement, it is not very secure for real-world use.

Why it is not secure:

1. Limited number of shifts:
    - There are only 25 possible shifts (1–25).
    - This means someone can try all of them quickly (brute force) and decrypt the message.
2. Letter frequency attacks:
    - Some letters appear more often in English (like 'E', 'T', 'A').
    - By looking at the frequency of letters, a person can guess the shift value and break the message.
3. Patterns are easy to see:
    - Repeating words or letters are shifted the same way, so patterns in the message remain.
    - This makes it easier to crack long messages.