## 

## 

## 

## Faculty of Computing

**Class: BSE-Section A&B 2025  
Course: Applications of ICT**

# **Lab 04: Exploring Networks with Wireshark**

**Date: 9/29/2025**

**Student Name: Tahreem Zafar**

**LMS I****D: 547588**

**Assembly Reflections:**

1. Registers are like tiny, super-fast storage spots inside the computer’s brain (the CPU). When you write Assembly code, you’re basically giving the CPU very direct orders to move data into these registers, do math or comparisons, and then put the results somewhere else The instructions themselves are simple but very specific t like “move this number here,” “add these two numbers,” or “jump to another part of the program.
2. Assembly is powerful and gives you total control, but it’s harder and slower to write. Python makes programming faster and simpler, letting you focus on the bigger picture instead of the tiny details

**Python Reflections :**

1. Python is easier because it takes care of a lot of the difficult things for you. Instead of worrying about things like managing memory, registers, or how the computer actually runs each instruction you just focus on the *logic* of what you want to do
2.  **Variables:** if you give names to data, so you don’t have to keep data of exact memory addresses or worry about the type of data at every step. You just say x = 5, and Python handles the rest.

 **Functions:** Help you group many codes into reusable blocks. Instead of writing the same instructions over and over, you write a function once and call it whenever you need it. This makes your code cleaner and easier to manage.

 **Loops:** Let you repeat actions without rewriting code multiple times. Whether it’s running through a list of items or doing something until a condition is met, loops make your programs shorter and more efficient.

**Comparison Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **feature** | **Assembly example** | **Python example** | **notes** |
| **Variable storage** | **Register(EAX)** | **X=5** | **Assembly uses AX,BX** |
| **Printing output** | **Int 21h** | **Print()** | **DOS INT 21h with AH=02h prints a char** |
| **Arithmetic** | **ADD AX,BX** | **x+y** | **Arithmetic is done using CPU instructions** |