National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Faculty of Computing

Class: BSE-Section A&B 2025

Course: Applications of ICT

Lab 03: Programming Concepts (Assembly and Python, Git/GitHub)

Date: 22, Sept 2025

Urooj Akmal

Objectives

By the end of this lab, students will:

1. Write a **mini project in Assembly** (low-level).
2. Write a **mini project in Python** (high-level).
3. Compare Assembly vs Python (low-level vs high-level abstraction).
4. Learn **Git/GitHub for versioning and collaboration**.
5. Work in **pairs** to experience real-world teamwork.

Part A: Assembly

Task: Build a **Mini Calculator (Addition & Multiplication)** in Assembly.

Requirements:

- Input two numbers (hard-coded in registers).
- Perform addition and multiplication.
- Display the results.

Deliverables:

- File: *`assembly_calc.asm`*
- Screenshot of output.

Part B: Python

Task: Build a **Feature-Rich Calculator** in Python.

Requirements:

1. Accept user input for two numbers.
2. Perform **+ – × ÷** operations.
3. Add an option to check if a number is even/odd.
4. Add an option to calculate percentage.
5. Use **functions** for each operation.
6. Use **loops** to let user perform multiple calculations until they exit.

Deliverables:

- File: *`python_calc.py`*
- Example run (screenshot).

Part C: Comparison Document (README.md)

Each group writes a **README.md** in their GitHub repo with:

1. **Assembly Reflections**
    - What did you notice about registers and instructions?
    - How is coding in Assembly different from Python?
2. **Python Reflections**
    - Why is Python easier/faster for building the same project?
    - Which features of Python help abstraction (variables, functions, loops)?
3. **Comparison Table**

Feature	Assembly Example	Python Example	Notes
Variable storage	Register (EAX)	x = 5	
Printing output	INT 21h	print()	
Arithmetic	ADD AX, BX	x + y	

Part D: Git/GitHub Collaboration

Students (working in **pairs**) will:

1. One student creates a repo *cs117-lab3*.
2. Add the partner as **collaborator** on GitHub.
3. Create folder structure:

```
lab3/
  assembly_calc.asm
  python_calc.py
  README.md
```

4. Both students:
    - Clone the repo.
    - Work on separate parts (one edits Assembly, one edits Python).
    - Commit and push changes.
    - Resolve merge conflicts if they happen.
5. Submit **repo link on LMS**.

Deliverables (LMS)

1. `assembly_calc.asm` (Assembly program).
2. `python_calc.py` (Python program).
3. `README.md` (comparison + reflection).
4. GitHub repo link.
