5.8 Final Module Assignment

I have selected two learning outcomes from the DEVP 600 - Business Innovation with Development Operations (DevOps) course, which is a first-year, second-semester course in our program's Cloud Computing and Blockchain structure.

LO 1: Explain the culture of development operations (DevOps).

Instructional Strategies:

- **Direct Instruction-Mini-Lecture + Concept Mapping**: A 30-minute interactive lecture introduces the three core DevOps values (collaboration, shared ownership, continuous improvement). Immediately afterward, students work in pairs to build a concept map—linking values to practices (e.g. DevOps cycles, cross-functional stand-ups). **Why it works:** The direct lecture gives precise terminology; the concept map forces learners to explain relationships in their own words, cementing comprehension (Comprehension/Summation).
- **Collaborative Learning-Scenario-Based Discussion**: Students are given a short case: "Team X's pipeline kept failing because developers and ops never spoke until post-mortem." In small groups, they discuss which DevOps cultural principles were missing and propose three low-cost fixes (e.g. daily stand-ups, shared dashboards). **Why it works:** By diagnosing a realistic failure, learners must apply and describe DevOps culture in context, reinforcing their ability to explain why each practice matters.

LO 2: Implement DevOps cloud automation.

Instructional Strategies:

Demonstration & Guided Practice-Live Demo + Guided Walkthrough: Instructor projects an AWS CodePipeline setup in Cloud9, walking through each step: from repo webhook to build, test and deploy stages. Students follow along in their own Cloud9 environments, executing each command. **Why it works:** Watching a live build-and-deploy sequence while simultaneously implementing in their own IDE solidifies the connection between abstract pipeline concepts and concrete automation tasks and applications.

Experiential Learning-Hands-On Lab with Peer Review: In a 90-minute lab, each student builds a simple CI/CD pipeline for a sample Node.js app on AWS. Afterward, they swap projects with a peer for code review, providing feedback on scripts, YAML config, and

deployment logs. **Why it works:** Building automation end-to-end ensures students implement real DevOps cloud practices; peer review promotes shared responsibility and continuous improvement, reflecting true DevOps culture even during technical work.

Reflective Practice & Application

- Reflection Journal: After each module, I'll write a brief entry answering: "What activity most improved students' understanding? Where did they struggle?" This habit will guide tweaks—e.g. breaking concept maps into smaller steps if mapping exercises prove too big.
- Student Feedback Survey: Mid-term, I'll deploy a 3-question survey ("Which activity helped you most with DevOps culture?", "What would you change about the pipeline lab?"). Sharing aggregated results with the class demonstrates I value their input and models a feedback loop.
- Peer Observation: I'll invite a colleague to observe the pipeline demo and lab, then compare notes on pacing and clarity, iterating on my live-coding scripts and scaffolding.

Key Takeaways from This Module

- **Phases of Curriculum Design Matter**
  I now appreciate the three distinct stages—planning, implementation, evaluation—that underlie effective curriculum development, rather than simply using existing course materials.
- **Outcome Alignment Is Fundamental**
  Revisiting Bloom's Taxonomy and the Learning Domains refocused my attention on crafting assessments and activities that genuinely support each learning outcome, not just delivering content.
- **Expand & Visualize Instructional Strategies**
  I discovered dozens of new delivery techniques beyond my usual repertoire. Posting a printed list of these strategies at home and on campus will serve as a constant prompt to experiment and diversify.
- **Make Reflection Tangible**
  Capturing my reflections—and inviting students to do the same in hard-copy journals—reinforces a feedback loop. Sharing those reflections with colleagues models continuous improvement and keeps learner needs front and center.