Engineering and Applied Science Programs for Professionals
Whiting School of Engineering
Johns Hopkins University
705.625 Agentic AI
Homework 3
Assigned with Module 9
Due at the end of Module 12

Total Points 100/100

Submission Instructions and Problem Selection:

While student discussions, sharing of recommendations, and some collaboration are encouraged for team building and peer learning, **each student is required to submit their own complete programming solutions.** Every submission must demonstrate the individual student's ability to design, document, implement in code, and test their work independently.

Grading Criteria (100% for each problem)

Completeness & Problem Coverage (20%): Implement all required parts (pseudocode, code, performance analysis, outputs) with nothing missing or vague.

Writing Quality, Technical Accuracy & Justification (20%): Explanations are clear, concise, and technically correct; design choices and conclusions are justified with sound reasoning.

Quantitative Work: Assumptions, Derivations & Calculations (20%): List assumptions up front; show derivations or intermediate formulas/metrics; present final results correctly and clearly.

Code Quality, Documentation & Execution (20%): The notebook runs end-to-end without errors; names are clear; formatting is consistent; comments explain key logic; the work is organized and reproducible.

Examples, Test Cases & Visuals (10%): Include labeled test cases and informative figures/tables with titles, captions, and axes; use appropriate metrics (e.g., precision, recall, F1, ROC/PR curves) rather than accuracy alone [1].

Notebook README & Reproducibility (10%): Provide Python version, package list with install steps, dataset details and download instructions, and step-by-step run instructions; use relative paths and fixed random seeds.

1. **Collaborative Multi-Agent Verification System — Modules 9 and 10**

   40 points total.

   Problem Statement:

   Scenario
   You have studied how multi-agent systems enable coordination, specialization, and critique across agents with distinct roles. In this assignment, you will design and implement a small team of collaborating agents that work together to answer and verify user queries **without retrieval augmentation**. The emphasis is on reasoning, validation, and system-level evaluation.

   Objective:
   Create a **multi-agent content verification system** that receives a user question or short prompt and returns a **final verified answer** after internal discussion, critique, and validation. Your system should demonstrate principles of **agency, rationality, communication, and evaluation**.

   System Requirements
   Implement your system using **LangGraph** (preferred) or another orchestration framework. Your system must include at least **three agents** with clearly defined and distinct roles.

Agent Role	Example Function
Orchestrator	Directs workflow, assigns subtasks, aggregates final result.
Generator / Responder	Produces an initial answer to the prompt.
Validator / Critic	Checks the response for factual soundness and coherence.
Refiner / Reviser	Revises the answer based on validator feedback.
Evaluator	Scores the system's own performance according to rubric.

   Functional Expectations

   (a) The system begins with a **user prompt** (e.g., a factual or conceptual question).

   (b) The **Generator Agent** produces an initial answer.

   (c) The **Validator Agent** critiques the answer (identifying errors, inconsistencies, or weak reasoning).

   (d) The **Refiner Agent** revises the answer based on validator feedback.

   (e) The **Orchestrator** collects and presents the final verified answer.

   (f) Each agent should exchange information through messages, maintaining reasoning transparency (e.g., via printed dialogue).

   Evaluation Component
   You must implement two complementary forms of evaluation:

   a. Automatic Metric (Quantitative) Choose a measurable indicator of improvement, such as:

   - Number of factual errors reduced.
   - String-match accuracy versus a reference answer.
   - Internal agreement score between agents.

   b. LLM-as-Judge or Rule-Based Rubric (Qualitative) Design a short rubric with 3–4 criteria (e.g., factual correctness, coherence, clarity, completeness). Use either a rubric-prompted LLM or deterministic rules to score the system's final output against the baseline (initial Generator answer). Include both evaluation results and a brief interpretation.

   Deliverables
   Submit two components:

(a) **Code Notebook or Script** (`.ipynb` or `.py`)

- Fully functional multi-agent system with printed dialogue or logs.
- Clear agent definitions and orchestration logic.

(b) **Summary of Methods (PDF or markdown/readme)**

- **Introduction:** Problem overview.
- **System Design:** Roles, architecture diagram, reasoning flow.
- **Evaluation Method:** Automatic + LLM-based metrics.
- **Results and Analysis:** Performance summary, what improved, what failed.
- **Reflection:** Insights about coordination, rationality, and evaluation.

Scoring Rubric (40 points total)

Category	Description	Points
System Design & Implementation	Functional multi-agent system (at least 3 roles), clear orchestration, logical communication flow.	15
Reasoning & Verification Mechanism	Evidence of critique–refine cycle; demonstrates rational coordination and bounded rationality.	10
Evaluation Framework	Includes both quantitative and qualitative assessments with clear metrics and interpretation.	8
Report Quality & Reflection	Clear writing, organization, visuals, and discussion of limitations and insights.	7

2. **Learning and Adaptation — Modules 7**

   20 points total (Each subpart is worth 10 points each).

   Problem Statement:

   Students should select **any two questions** from Questions 1–6 in the provided Jupyter notebook titled `705625_Assignment07_v2.ipynb`. Each selected question is worth **10 points**. All responses and code must be written directly in the notebook cells provided.

   Be sure to **download and extract the accompanying photos ZIP file** before opening the notebook so that all embedded images render correctly. The completed notebook (`705625_Assignment07_v2.ipynb`) will be submitted as part of your final submission package.

3. **Learning and Adaptation — Module 7**

   40 points total.

   Problem Statement:

   Complete **Question 7** in the same Jupyter notebook (`705625_Assignment07_v2.ipynb`). This question is worth **40 points** and should be answered directly within the notebook cells provided.

   Ensure that all required outputs, figures, and written responses are visible and properly formatted before submission. The updated notebook file will be included as part of your overall assignment submission.