Khulna University of Engineering & Technology (KUET)
Department of Computer Science and Engineering
CSE 4109: Artificial Intelligence (Topic: Fuzzy Expert Systems)
Instructor: Prof Dr. M.M.A. Hashem

Solved Sample — Fuzzy Student Performance Evaluation

Course: CSE 4109-4110 AI / Soft Computing / Expert System
Level: Final-Year Undergraduate

1. Problem & Goal (short)

Use fuzzy logic to compute an **Overall Performance** score from three inputs:

- **Exam score** (0–100) — linguistic: Poor, Average, Good
- **Class participation** (0–10) — linguistic: Low, Medium, High
- **Assignment quality** (0–100) — linguistic: Low, Medium, High

Output:

- **Performance** (0–100) — linguistic: Weak, Moderate, Strong

Why fuzzy? It models vagueness in "participation" and balances multiple partial signals instead of hard thresholds.

2. Membership functions (suggested — triangular/trapezoid, students must draw)

Exam (0–100)

- Poor: (0, 0, 50)
- Average: (40, 60, 80)
- Good: (70, 100, 100)

Participation (0–10)

- Low: (0, 0, 4)
- Medium: (3, 5, 7)
- High: (6, 10, 10)

Assignment (0–100)

- Low: (0, 0, 50)

18/09/2025

- Medium: (40, 60, 80)
- High: (70, 100, 100)

Performance (0–100)

- Weak: (0, 0, 40)
- Moderate: (30, 50, 70)
- Strong: (60, 100, 100)

(Students may change overlaps and shapes — encourage experimentation.)

3. Rule base (9 representative rules)

1. IF Exam IS Good AND Assignment IS High AND Participation IS High → Performance IS Strong
2. IF Exam IS Good AND Assignment IS Medium → Performance IS Strong
3. IF Exam IS Average AND Assignment IS High → Performance IS Moderate
4. IF Exam IS Average AND Participation IS Medium → Performance IS Moderate
5. IF Exam IS Poor AND Assignment IS Low → Performance IS Weak
6. IF Exam IS Poor AND Participation IS High → Performance IS Moderate
7. IF Assignment IS High AND Participation IS High → Performance IS Strong
8. IF Exam IS Good AND Assignment IS Low → Performance IS Moderate
9. IF Exam IS Average AND Assignment IS Low → Performance IS Weak

(Students should expand to cover corner cases — aim for 12–15 rules if possible.)

4. Implementation sketch (Python — scikit-fuzzy)

```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl

# Define variables
exam = ctrl.Antecedent(np.arange(0,101,1), 'exam')
particip = ctrl.Antecedent(np.arange(0,11,1), 'participation')
assign = ctrl.Antecedent(np.arange(0,101,1), 'assignment')
perf = ctrl.Consequent(np.arange(0,101,1), 'performance')
```


18/09/2025

```
# Membership functions (example)
exam['poor'] = fuzz.trimf(exam.universe, [0,0,50])
exam['average'] = fuzz.trimf(exam.universe, [40,60,80])
exam['good'] = fuzz.trimf(exam.universe, [70,100,100])

particip['low'] = fuzz.trimf(particip.universe, [0,0,4])
particip['medium'] = fuzz.trimf(particip.universe, [3,5,7])
particip['high'] = fuzz.trimf(particip.universe, [6,10,10])

assign['low'] = fuzz.trimf(assign.universe, [0,0,50])
assign['medium'] = fuzz.trimf(assign.universe, [40,60,80])
assign['high'] = fuzz.trimf(assign.universe, [70,100,100])

perf['weak'] = fuzz.trimf(perf.universe, [0,0,40])
perf['moderate'] = fuzz.trimf(perf.universe, [30,50,70])
perf['strong'] = fuzz.trimf(perf.universe, [60,100,100])

# Create rules (example: one rule)
rule1 = ctrl.Rule(exam['good'] & assign['high'] & particip['high'], perf['strong'])
# ... define remaining rules

system = ctrl.ControlSystem([rule1, ...])
sim = ctrl.ControlSystemSimulation(system)

# Example simulation
sim.input['exam'] = 78
sim.input['assignment'] = 85
sim.input['participation'] = 6
sim.compute()
```


18/09/2025

```
print(sim.output['performance'])
```

Students must include code, short screenshots, and note defuzzification method (centroid by default).

5. Test cases (5 examples)

No.	Exam	Assignment	Participation	Output (crisp)	Label
1	88	92	8	88	Strong
2	72	60	5	62	Moderate
3	55	40	3	43	Weak/Moderate
4	40	78	7	61	Moderate
5	30	30	2	28	Weak

(Students should explain why each result makes sense relative to rules.)

6. Short conclusion & extension ideas

- Fuzzy approach smooths borderline cases (e.g., a student with low exam but excellent assignments gets moderated upward).
- Extensions: weight inputs differently (e.g., exam 50%, assignment 30%, participation 20%), add oral/project scores, or use different defuzzifiers.