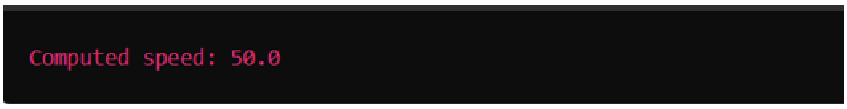
FUZZY INFERENCE SYSTEM

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

# Create fuzzy variables
distance = ctrl.Antecedent(np.arange(0, 11, 1), 'distance')
speed = ctrl.Consequent(np.arange(0, 101, 1), 'speed')

# Define membership functions for distance
distance['near'] = fuzz.trimf(distance.universe, [0, 0, 5])
distance['medium'] = fuzz.trimf(distance.universe, [0, 5, 10])
distance['far'] = fuzz.trimf(distance.universe, [5, 10, 10])

# Define membership functions for speed
speed['slow'] = fuzz.trimf(speed.universe, [0, 0, 50])
speed['medium'] = fuzz.trimf(speed.universe, [0, 50, 100])
speed['fast'] = fuzz.trimf(speed.universe, [50, 100, 100])

# Define rules
rule1 = ctrl.Rule(distance['near'], speed['slow'])
rule2 = ctrl.Rule(distance['medium'], speed['medium'])
rule3 = ctrl.Rule(distance['far'], speed['fast'])

# Create the control system
speed_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
car_speed = ctrl.ControlSystemSimulation(speed_ctrl)

# Input distance and compute speed
car_speed.input['distance'] = 7
car_speed.compute()
```

```
# Print the computed speed

print("Computed speed:", car_speed.output['speed'])
```

In this example, we first define the fuzzy variables `distance` and `speed` using the `Antecedent` and `Consequent` classes, respectively. We then define the membership functions for each variable using `fuzz.trimf`. Next, we define the rules that determine the speed based on the distance using the `ctrl.Rule` class. After defining the rules, we create the control system using `ctrl.ControlSystem` and `ctrl.ControlSystemSimulation`. Finally, we input a distance value and compute the speed using `compute()`, and print the computed speed.

This is a simple example to demonstrate the basic structure of a fuzzy inference system in Python using `scikit-fuzzy`. You can expand on this example by adding more variables, membership functions, and rules to create a more complex fuzzy inference system for your specific application.



```
Computed speed: 50.0
```