Task Description:

You are tasked with creating a Python program that uses fuzzy logic to assess the risk level of heart disease based on input parameters like age, sex, chest pain type (cp), resting blood pressure (trestbps), cholesterol level (chol), fasting blood sugar (fbs), resting ECG results (restecg), maximum heart rate achieved (thalach), exercise-induced angina (exang), oldpeak (ST depression induced by exercise), slope of the peak exercise ST segment (slope), number of major vessels colored by fluoroscopy (ca), and thalassemia type (thal). The system should classify the risk as Low, Medium, or High based on fuzzy rules.

The implementation should use the `skfuzzy` library to define membership functions, create fuzzy rules, and perform inference.

Deliverables:

1. **Python Script 1 (`heart_disease_assessment.py`)**:
   - Handles user interaction.
   - Accepts input values for age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, and thal.
   - Calls functions from the second script (`fuzzy_logic.py`) to compute the heart disease risk level.
   - Displays the fuzzy classification result.
2. **Python Script 2 (`fuzzy_logic.py`)**:
   - Implements fuzzy logic functions.
   - Defines membership functions for age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, and thal.
   - Establishes fuzzy rules.
   - Uses fuzzy inference to determine the heart disease risk level.
3. **Utlis Script (`utils.py`)**:
   - Contains helper functions for data validation and formatting results such covering ranges validation.

Requirements List:

1. **Fuzzy Logic Implementation:**
   - Use the `skfuzzy` library to define fuzzy sets.
   - Implement fuzzy membership functions for input variables (age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal).
   - Define fuzzy rules linking input variables to the risk level.
   - Use the Mamdani inference system to determine the final classification.
2. **Fuzzy Membership Functions:**
   - Define fuzzy sets (e.g., Low, Medium, High) for each input variable.

    - Use Gaussian or triangular membership functions.

3. **Fuzzy Rules:**
    - Define rules such as:
        - "If cholesterol is high and blood pressure is high, then risk is high."
        - "If chest pain type is severe and thalach is low, then risk is high."
        - "If age is low and thalach is high, then risk is low."
        - "If oldpeak is high and slope is downward, then risk is high."
4. **Fuzzy Inference System:**
    - Apply fuzzy inference to compute the final heart disease risk level.
    - Use centroid defuzzification to convert fuzzy output into a crisp value.
5. **Error Handling:**
    - Validate user input to ensure values are within expected ranges.
    - Handle edge cases like missing or incorrect inputs.
6. **Functions to Implement:**
    - `compute_risk(age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal)`: Uses fuzzy logic to classify risk.
    - `define_membership_functions()`: Creates fuzzy membership functions.
    - `define_fuzzy_rules()`: Establishes the rule base.
    - `defuzzify_output(fuzzy_result)`: Converts fuzzy output into a numerical risk score.

File Structure:

```
fuzzy_heart_disease/
 ├── heart_disease_assessment.py  # Handles user interaction
 ├── fuzzy_logic.py               # Contains fuzzy logic functions
 └── utils.py                     # Contains helper functions (e.g., validation)
```

Example User Test:

```
Enter age: 55
Enter sex (0 = female, 1 = male): 1
Enter chest pain type (0-3): 2
Enter resting blood pressure: 140
Enter cholesterol level: 250
Enter fasting blood sugar (0 = False, 1 = True): 0
Enter resting ECG results (0-2): 1
Enter maximum heart rate achieved: 150
Enter exercise-induced angina (0 = No, 1 = Yes): 0
Enter oldpeak: 1.5
Enter slope (0-2): 2
Enter number of major vessels (0-3): 1
Enter thalassemia type (0-3): 2
Heart Disease Risk Level: Medium
```

This task ensures a structured implementation of fuzzy logic while maintaining modularity and clarity.