

Q. Write a program to demonstrate the membership function calculation in fuzzy logic. You can use any fuzzifier.

To implement Fuzzy Logic in Java program, a library called jFuzzyLogic is used. jFuzzyLogic is an open source fuzzy logic library to simplify fuzzy systems developments. jFuzzyLogic is a fuzzy logic package and it is written in Java. jFuzzyLogic implements Fuzzy control language (FCL) specification IEC 61131 part 7.

Citations:

1. Cingolani, Pablo, and Jesús AlcaláFdez. "jFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming" (pdf/Cingolani_AlcalaFdez_jFuzzyLogic_2013_IJCIS.pdf)

2. Cingolani, Pablo, and Jesus AlcalaFdez. "jFuzzyLogic: a robust and flexible FuzzyLogic inference system language implementation." (pdf/jFuzzyLogic.pdf) Fuzzy Systems (FUZZIEEE), 2012 IEEE International Conference on. IEEE, 2012.

Implementation Example:

In this example, a fuzzy system is implemented in a bathtub where the cold water valve or the hot water valve is controlled automatically as per the temperature of the environment. For this demonstration, a library called jFuzzyLogic is used and is coded in Java language. To define the parameters for the fuzzification and defuzzification rule, Fuzzy control language (FCL) is implemented which has to be written separately and should be implemented in code with the help of jFuzzyLogic library.

Coding:

bathtub.fcl

```
FUNCTION_BLOCK TemperatureControl    // Block definition

VAR_INPUT                            // Define input variables
    temp: REAL;
END_VAR

VAR_OUTPUT                            // Define output variables
    hot_valve: REAL;
    cold_valve: REAL;
END_VAR

FUZZIFY temp                          // Fuzzify input variable 'temp': {'hot', 'cold'}
    TERM hot    :=    (35,0)(90,1);
    TERM cold   :=    (1,1)(20,1)(50,0);
END_FUZZIFY

DEFUZZIFY hot_valve                  // Defuzzify output variable 'hot_valve' : {'open', 'closed'}
    TERM open   :=    100;
    TERM closed :=    0;
    METHOD : COGS;
    DEFAULT    :=    0;
END_DEFUZZIFY
```

```

DEFUZZIFY cold_valve      // Defuzzify output variable 'cold_valve' : {'open', 'closed'}
    TERM open      :=      100;
    TERM closed    :=      0;
    METHOD : COGS;      // Use 'Center Of Gravity' defuzzification method
    DEFAULT := 0;      // Default value is 0 (if no rule activates defuzzifier)
END_DEFUZZIFY

RULEBLOCK Control
    AND : MIN;          // Use 'min' for 'and' to Fulfill DeMorgan's law
    ACT : MIN;          // Use 'min' activation method
    ACCU : MAX;         // Use 'max' accumulation method

    //Define the rules to be used
    RULE 1:      IF temp IS cold THEN hot_valve IS open;
    RULE 2: IF temp IS hot THEN cold_valve IS open;
END_RULEBLOCK

END_FUNCTION_BLOCK

```

TempControl.java

```

import net.sourceforge.jFuzzyLogic.FIS;
import net.sourceforge.jFuzzyLogic.FunctionBlock;
import net.sourceforge.jFuzzyLogic.plot.JFuzzyChart;
import net.sourceforge.jFuzzyLogic.rule.Variable;
import java.util.Scanner;

public class TempControl {
    public static void main(String[] args) throws Exception {
        String filename = "bathtub.fcl";
        FIS fis = FIS.load(filename, true);
        if (fis == null) {
            System.err.println("Can't load file: '" + filename + "'");
            System.exit(1);
        }
        FunctionBlock fb = fis.getFunctionBlock(null);

        System.out.println("Enter the temperature.");
        Scanner tempScanner = new Scanner(System.in);
        double temp = tempScanner.nextDouble();
        fb.setVariable("temp", temp);

        fb.evaluate();

        fb.getVariable("hot_valve").defuzzify();
        fb.getVariable("cold_valve").defuzzify();
        JFuzzyChart.get().chart(fb);

        Variable hot_valve = fb.getVariable("hot_valve");
        Variable cold_valve = fb.getVariable("cold_valve");
        JFuzzyChart.get().chart(hot_valve, hot_valve.getDefuzzifier(), true);
        JFuzzyChart.get().chart(cold_valve, cold_valve.getDefuzzifier(), true);

        double hotVal = fb.getVariable("hot_valve").getValue();
        double coldVal = fb.getVariable("cold_valve").getValue();

        if(hotVal == 100 && coldVal == 0){
            System.out.println("It's cold outside, Open the HOT VALVE");
        }
        if(coldVal == 100 && hotVal == 0 ){

```

```

        System.out.println("It's hot outside, open the COLD VALVE");
    }
    if(hotVal == coldVal){
        System.out.println("It's neither cold nor hot outside. So, any valve can be
opened.");
    }
}
}
}

```

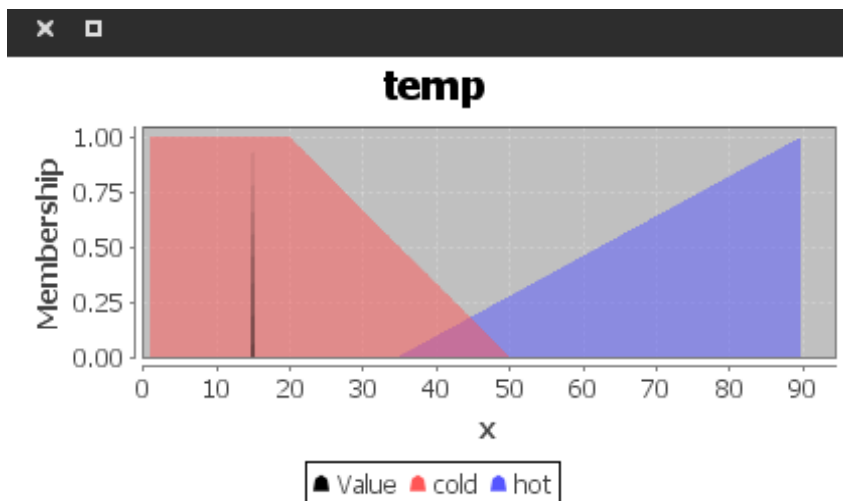
OUTPUT 1:

```

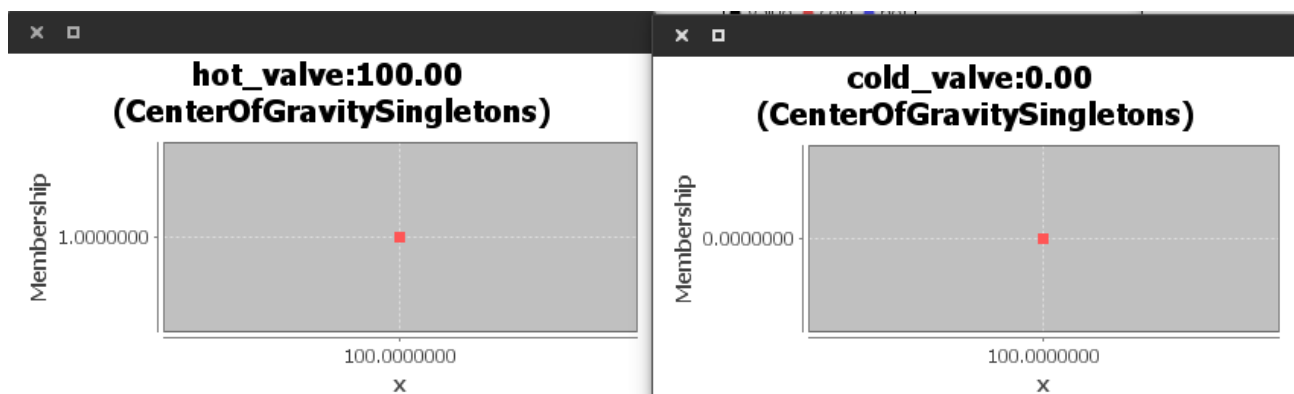
/usr/local/java/jdk1.7.0_55/bin/java ...
Enter the temperature.
15
It's cold outside, Open the HOT VALVE

```

Input & output in CLI



Fuzzification with input



Outputs indicating degree of membership

OUTPUT 2:

```
/usr/local/java/jdk1.7.0_55/bin/java ...  
Enter the temperature.  
50  
It's hot outside, open the COLD VALVE  
|
```

OUTPUT 3:

```
/usr/local/java/jdk1.7.0_55/bin/java ...  
Enter the temperature.  
40  
It's neither cold nor hot outside. So, any valve can be opened.  
|
```