Programming Assignment 3-1

1. Problem Definition

The code needs to be able to create three numbers at random, list them out, then pick the lowest value of them and print that one by itself. The challenge will be to create an efficient “if” statement.

2. Input and Output.

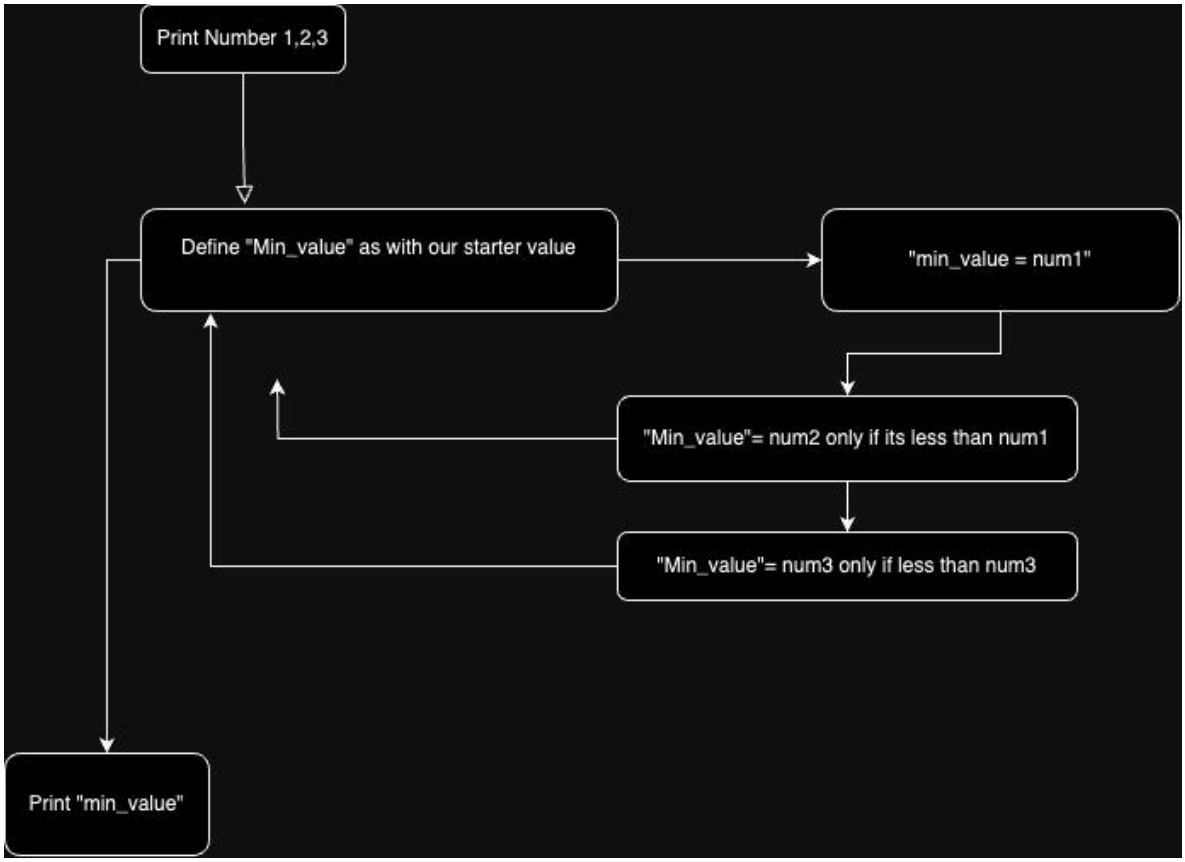
- Input: are three numbers randomly generated
- Output: List of the three numbers generated, then the lowest number printed by itself

3. Variables

- Number1 = random.randint(0,100)
- Number2 = random.randint(0,100)
- Number3 = random.randint(0,100)
- min_value

4. Flow Chart

**** Flow chart Tips. In your VS code, install **drawio** Extension. And then create a file “hello.drawio”. You can edit the flowchart in VS Code. Once you complete the flow chart, **export** it as a **PNG** file. And commit/push to GitHub Repository together.
This is the same interface as the website https://draw.io



5. Elaboration on Algorithms

Algorithm Steps:

The algorithm to find the minimum value among three integers can be described in the following steps:

1. Consider the three integer values as inputs: number1, number2, and number3.
2. Set a variable, let's call it "***min_value***," to store the minimum value.
3. Initially, assign the value of ***number1*** to ***minValue***.
    - minvalue = number1
4. Compare the value of number2 with ***minValue***. If number2 is less than minValue, update minValue to be number2. If not keep it as number1

5. Compare the value of number3 with ***minValue***. If Number3 is less than minValue, update minValue to be Number3. If not, keep it to either number1, or number2 depending on which one is less.
6. At this point, ***minValue*** will hold the smallest value among the three integers.

Explanation:

The algorithm starts by assigning the value of the first integer, 'number1,' to the variable 'minValue.'

Next, the algorithm compares the value of 'number1' with 'minValue.' If 'number2' is smaller than 'minValue,' the algorithm updates 'minValue' to hold the value of 'number2.'. If 'number2' had a bigger value, then the program would result back to 'number1'.

The algorithm then compares the value of 'Number2' with 'minValue.' If 'number3' is smaller than 'minValue,' the algorithm updates 'minValue' to hold the value of 'Number3'. If 'number. If 'number3' had a bigger value, then it would result back to 'number2' then that would result back to 'number1'.

This step ensures that the smallest value encountered so far is stored in 'minValue.'

By doing so, the algorithm ensures that 'minValue' contains the smallest value among 'number1,' 'number2,' and 'number3.'

Conclusion:

In conclusion, the algorithm to find the minimum value among three integer values involves comparing the values and updating a variable, 'minValue,' accordingly.

The algorithm efficiently determines the smallest value by sequentially comparing the integers and updating 'minValue' when a smaller value is encountered.

6. Errors and Lessons

- Finding the right 'if' statements
    - Creating the 'A' < 'B', "B<A, etc. The program statements to make the program at least pick the smallest number'
- Indentation errors

    - else and elif were giving me a hard time when it came to formatting.

- Lessons:
    - I learned about the “if” statements being able to be paired with "else in the same line. This helps create a more organized look and it's easier to fix issues. I also learned a great way to be able to grab the least valued number from the code in a really effective way. Also finding out about ‘Random.randint’ is a cool line of code I hope to be able to implement in my work.