PrimeNumber Part:B Mastery ReflectionLog

The majority of the code is identical to the Part A of this project, the primary difference being that the modulo calculation takes place in the isPrime method.

The main method is similar to Part A but with the boolean, and if-else removed, along with the count declared outside of the while loop. The loop continues to increment the count by 1 while it is between 2, and one less than the number. The if-else checks if the isPrime method returns true, or if the number is one, like the original.

The isPrime method returns a boolean value by checking if the modulo of the number by the count returns zero.

```
package Mastery;

import java.util.Scanner;

public class PrimeNumbersPartB {

    public static boolean isPrime(double number, double count) {

        //Method returns true if the number has no remainder when divided
        return number % count == 0;

    }

    public static void main(String[] args) {

        //Preparing for user input
        Scanner userInput = new Scanner(System.in);

        //Prompt and record the user for
        System.out.print("Enter an integer value: ");
        double number = userInput.nextInt();

        //Declaration
        double count= 2;

        //Increments the divisor by one, while 2 less than the inputed number and if the modulo != 0.
        while(count <= (number - 2) && isPrime(number, count) != true) {
            count++;
        }

        //Checks if the method returns true or the number is one, and prints appropriate response
        if (isPrime(number, count) || number == 1) {
            System.out.print("Isn't prime");
        } else {
            System.out.print("Is prime");
        }

    }

}
```