

# PrimeNumber Part:A Mastery ReflectionLog

This is the first rendition of my code. Originally I had planned to do everything within a single if else statement within a loop. This loop would run between while the inputted number was being divided by 2, and one less than the inputted number. If the modulo of the number would equal zero then I would break the statement and declare it as non-prime.

```
package Mastery;

import java.util.Scanner;

public class PrimeNumberPartA {

    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: ");
        int number = userInput.nextInt();

        for (int count=2; count <= (number - 1); count++) {
            if (number % count == 0) {
                System.out.print("Number isn't prime");
                break;
            } else {
            }
        }
    }
}
```

I then added functionality for prime declaration within the same if else statement. Because the else statement also broke the loop it resulted in any odd numbers being declared as prime.

```

package Mastery;

import java.util.Scanner;

public class PrimeNumberPartA {

    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: ");
        int number = userInput.nextInt();

        for (int count=2; count <= (number - 1); count++) {
            if (number % count == 0) {
                System.out.print("Number isn't prime");
                break;
            } else {
                System.out.print("Number is prime");
                break;
            }
        }
    }
}

```

---

I fixed this by removing the break from the else portion, and added a boolean value that would be set to false by default, and would be set true if the modulo ever resulted in non-zero. Unfortunately this meant that 2 would return as non-prime. Additionally I moved the print statement outside of the original if else statement to an if else if.

```

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

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

boolean prime = false;

for (int count=2; count <= (number - 1); count++) {
    if (number % count == 0) {
        prime = false;
        break;
    } else {
        prime = true;
    }
}

if (prime == false) {
    System.out.print("Number isn't prime");
} else if (prime == true) {
    System.out.print("Number is prime");
}

```

I then swapped the boolean to be true by default, this meant that both 2 and 1 would return as prime. I changed the final if-else-if to an if-else and added an additional or statement so that 1 would return as non-prime.

```

//Declaration
boolean prime = true;

//Loops an if else statement and increments count by 1 while the
//count is less than or equal to 2 less than the inputed number
for (int count=2; count <= (number - 2); count++) {
    //Checks if the modulo of the number is zero
    if (number % count == 0) {
        //Declares the number as not prime and breaks loop
        prime = false;
        break;
    } else {
        //Declares it as prime
        prime = true;
    }
}

```

If-else statement with or:

```
//Checks if the number is prime or is 1
if (prime == false || number == 1) {
    //Prints according statement
    System.out.print("Number isn't prime");
} else {
    System.out.print("Number is prime");
}
```

Finally I decided to clean up the code by removing the needless else statement when checking if the number was prime.

```
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: ");
    int number = userInput.nextInt();

    //Declaration
    boolean prime = true;

    //Loops an if else statement and increments count by 1 while the
    //count is less than or equal to 2 less than the inputed number
    for (int count=2; count <= (number - 2); count++) {
        //Checks if the modulo of the number is zero
        if (number % count == 0) {
            //Declares the number as not prime and breaks loop
            prime = false;
            break;
        }
    }

    //Checks if the number is prime or is 1
    if (prime == false || number == 1) {
        //Prints according statement
        System.out.print("Number isn't prime");
    } else {
        System.out.print("Number is prime");
    }
}
```