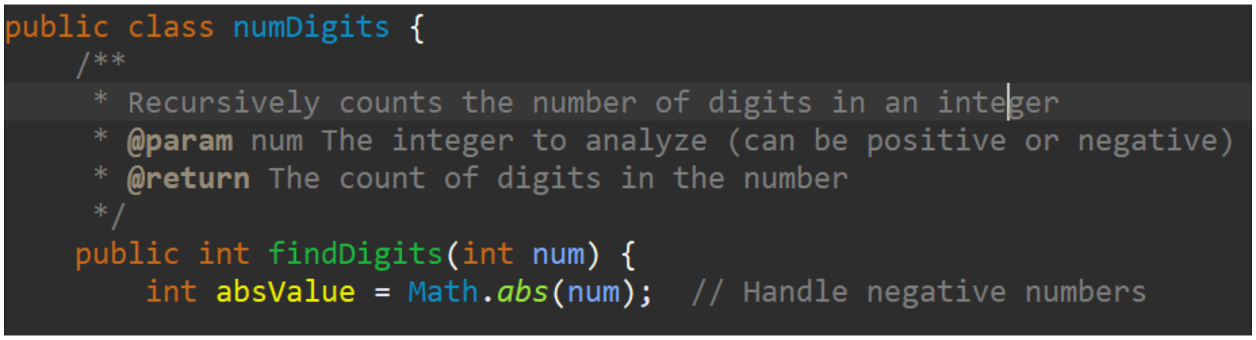
Credit Name: CSE 3110 Iterative Algorithms 1

Assignment: numDigits

How has your program changed from planning to coding to now? Please Explain

For this program, I have designated 2 classes: one that recursively determines the number of digits for a given integer, and the other which allows the user to test the recursion method for their choice of integer.

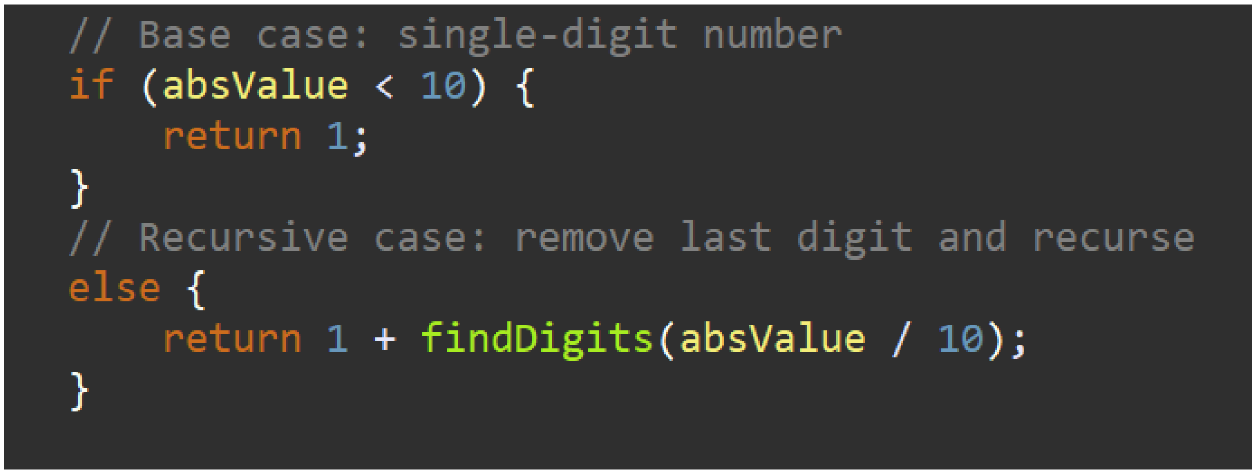
Firstly, I started with the numDigit class which defines the method that calculates the total number of digits.



```java
public class numDigits {
    /**
     * Recursively counts the number of digits in an integer
     * @param num The integer to analyze (can be positive or negative)
     * @return The count of digits in the number
     */
    public int findDigits(int num) {
        int absValue = Math.abs(num);  // Handle negative numbers
```

In this method, there is one integer parameter which is the number that we will find the digit count for. The only limitation of this method is the space allocated for int variables, really large numbers will not work (above 2147483647).

Firstly, we need to find the absolute value of the digit so that negative integers work as well. This is done using the abs method in the Math class from java.



```java
        // Base case: single-digit number
        if (absValue < 10) {
            return 1;
        }
        // Recursive case: remove last digit and recurse
        else {
            return 1 + findDigits(absValue / 10);
        }
```

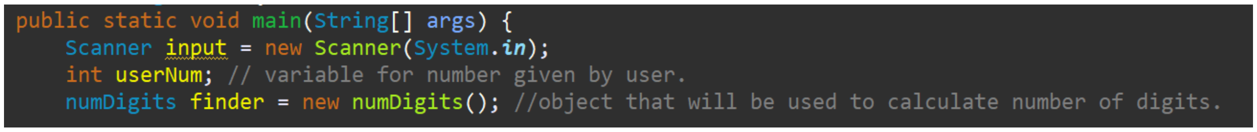
Then, we have an if-else statement which has 2 cases: a base case that returns 1 when the absolute value is less than 10. The logic behind this is that if a number is below 10 it

is 0-9 which has 1 digit. If the number is larger than 10, it enters a recursive call with the original absolute value divided by 10. This recursive call keeps occurring until the absValue is less than 10 (i.e. we reach the last digit). An example of how this method runs is as follows:

If num = -1000:
The absValue = 1000.
1000 satisfies the else statement, so it goes into a recursive call and is waiting for a value to be returned for findDigits(absValue/10) .

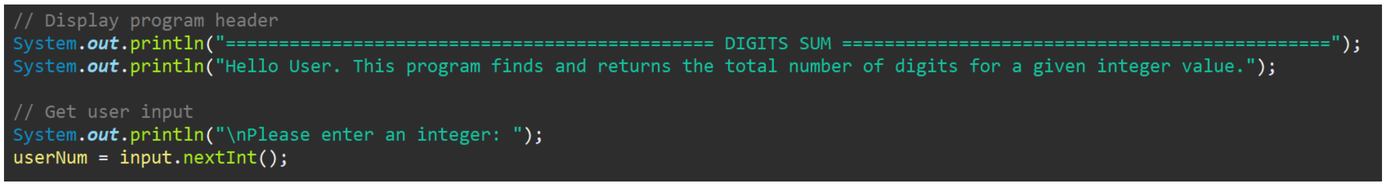
Now the method is called again with 100 instead. Again it goes into the else and the method recursively calls 10. Then it recurses one last time and we get 1. 1 satisfies the if condition and it returns 1. 1 is returned as the value for findDigits(10/10) which adds with the 1 to give 2 for 10. Then 2 is returned as the value for findDigits(100/10) and adds with the 1 in the return statement to give 3. The same thing happenes one last time for findDigits(1000/10) and returns 4 after 1 is added. Thus 4 is returned at the end.

Now that this method works, we can make a tester class to allow the user to test with a plethora of options.



```java
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    int userNum; // variable for number given by user.
    numDigits finder = new numDigits(); //object that will be used to calculate number of digits.
```

First we define this class to execute our main method. Then we declare and initialize both a scanner object for user input and our numDigits object defined in the previous class. We have declared an integer variable userNum for the value given by the user.

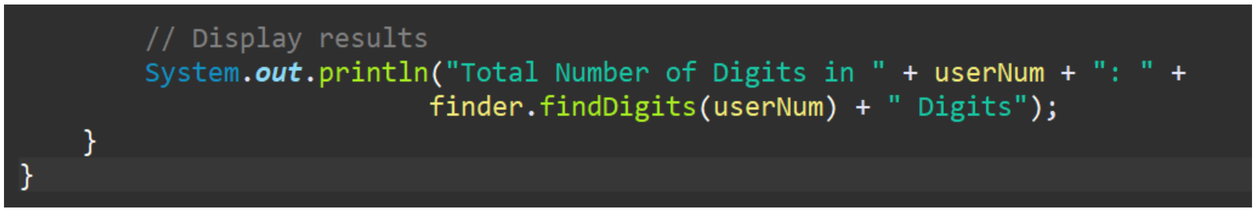


```java
// Display program header
System.out.println("============================================== DIGITS SUM ==============================================");
System.out.println("Hello User. This program finds and returns the total number of digits for a given integer value.");

// Get user input
System.out.println("\nPlease enter an integer: ");
userNum = input.nextInt();
```

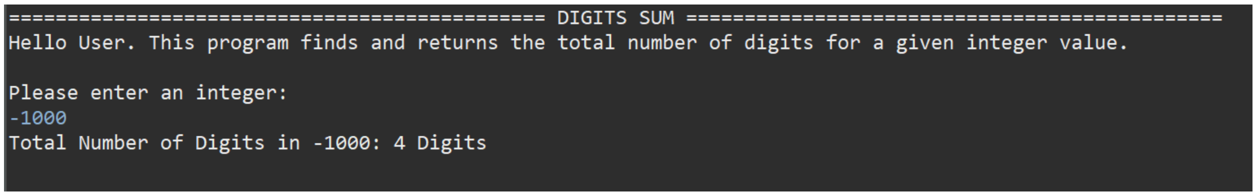
Then we print an overview of how this program works because the title alone can be misinterpreted.

We then prompt and receive an integer from the user.



```java
        // Display results
        System.out.println("Total Number of Digits in " + userNum + ": " +
                            finder.findDigits(userNum) + " Digits");
    }
}
```

Lastly we print a statement which shows the number of digits in the users number by calling the findDigits method in our finder object.



```
============================================== DIGITS SUM ==============================================
Hello User. This program finds and returns the total number of digits for a given integer value.

Please enter an integer:
-1000
Total Number of Digits in -1000: 4 Digits
```

This is an example output for -1000.