DigitExtractor ReflectionLogs -

Started off by asking the user to input a three-digit number and then showing them the possible choices they can make in regards to the number they declared. Variable “choice” will record the user’s input based on what they choose.

```
package Mastery;
import java.util.*;

public class DigitExtractorClient {

    public static void main(String[] args) {

        DigitExtractor CoolObject =  new DigitExtractor();// Object here
        Scanner Input = new Scanner(System.in);
        System.out.print("Enter a three-digit number - ");
        int number = Input.nextInt();
        System.out.println("Show the (N)umber");
        System.out.println("Show the (O)nes place number");
        System.out.println("Show the (T)ens place number");
        System.out.println("Show the (H)undreds place number");
        System.out.println("(Q)uit");
        String choice;
```

Following that, I created a function in the other file, which will record the value of the integer entered by the user

```
public class DigitExtractor {

    private int number;

    /*
     * Sets value of the number as the input.
     * pre: none
     * post: Value of number has been set.
     */
     public void setNumber(int i) {
     number = i; }
```

After that was set up, I created a few other functions which would return the values of digits at each place (hundreds, tens, ones) in the original integer

```
/*
 * Returns the digit at hundreds place.
 * pre: none
 * post: The hundreds place digit has been returned.
 */
 public int hundreds() {
 int hundreds = number/100;
 return(hundreds); }

 /*
 * Returns the digit at tens place.
 * pre: none
 * post: The tens place digit has been returned.
 */
 public int tens() {
 int temp1 = number%100;
 int tens = temp1/10;
 return(tens); }

 /*
 * Returns the digit at ones place.
 * pre: none
 * post: The ones place digit has been returned.
 */
 public int ones() {
 int temp1 = number%100;
 int ones = temp1%10;
 return(ones); }
```

Back to the client code, I gave the inputted number value to the setNumber() function, which enabled the rest of the functions to calculate their numbers and return them based on the choice that the user makes. The do-while loop ensures that it runs once at least, and continues working until the user enters Q.

```
CoolObject.setNumber(number);

do {
    System.out.print("Enter your choice - ");
    choice = Input.next();

    if (choice.equalsIgnoreCase("N")) {
        System.out.println(number);
    }
    else if (choice.equalsIgnoreCase("O")) {
        System.out.println(CoolObject.ones());
    }
    else if (choice.equalsIgnoreCase("T")) {
        System.out.println(CoolObject.tens());
    }
    else if (choice.equalsIgnoreCase("H")) {
        System.out.println(CoolObject.hundreds());
    }
    else {
        break;
    }
} while (choice != "Q" || choice != "q");
```