```java
/**
 * Main
 */
public class Main {

    /**
     * main()
     *
     * checkRangeForPrime(int rangeFrom, int rangeTo)
     * isPrime(int n)
     * calculateInterest(double amount, double interestRate)
     *
     * @param args args
     */
    public static void main(String[] args) {

        /*
         * for-loop intro
         */
        // interest = Zins
        // interest rate = Zinssatz
        System.out.println("$10'000 at 2% = $" + calculateInterest(10_000.0, 2.0) + " $");

        for(int i=0; i<5; i++) {
            System.out.println("Loop: " + i); // Loop: 0 > Loop: 1 > Loop: 2 > ...
        }
        System.out.println();

        for(int i=0; i<=10; i+=2) {
            System.out.println("Even number: " + i); // Even number: 0 > Even number: 2 > Even numbers: 4 > ...
        }
        System.out.println();


        /*
         * Challenge 0
         *
         * NEW: String.format(String format, Object... args)
         */
        // using the for statement, call the calculateInterest method with
        // the amount of 10000 with an interestRate of 2,3,4,5,6,7, and 8
        // and print the results to the console window.
        for(int i=2; i<9; i++) {    // i auto-converted to double by Java

            // output @ 7% interest rate:
            // $10'000 at 7% interest rate = $700.0000000000001
            System.out.println("$10'000 at " + i + "% interest rate = $" + calculateInterest(10_000.0, i));

            // String.format(String format, Object... args) = pass the value form the method trough the String.format command.
            // "%.2f" = converts the number and output with 2 decimal points.
            // output @ 7% interest rate:
            // $10'000 at 7% interest rate (formatted String) = $700,00
            double interest = calculateInterest(10_000.0, i);
            String formattedInterest = String.format("%.2f", interest); // String.format(String format, Object... args)
            System.out.println("$10'000 at " + i + "% interest rate (formatted String) = $" + formattedInterest);
        }

        System.out.println("******************");
        System.out.println();


        /*
         * Challenge 1
         */
        // How would you modify the for loop above to do the same thing as
        // shown but to start from 8% and work back to 2%
        for(int i=8; i>1; i--) {
            // output @ 7% interest rate:
            // $10'000 at 7% interest rate (formatted String) = $700,00
            System.out.println("$10'000 at " + i + "% interest rate = $" + String.format("%.2f", calculateInterest(10_000.0, i)));
        }
        System.out.println();


        /*
         * Challenge 2:
         * Check range for prime.
         */
        checkRangeForPrime(0, 100);
    }


    /**
     * checkRangeForPrime()
     * Checks a range of numbers for primes.
     */
    // Create a for statement using any range of numbers
    // Determine if the number is a prime number using the isPrime method
    // if it is a prime number, print it out AND increment a count of the
    // number of prime numbers found
    // if that count is 3 exit the for loop
    // hint:  Use the break; statement to exit
    public static void checkRangeForPrime(int rangeFrom, int rangeTo) {

        int primeCounter = 0; // amount of primes found within the range
        int maxNumberOfPrimesToFind = 10; // abort calculation after max. number of primes

```

```java
        for(int i = rangeFrom; i <= rangeTo; i++) {

            boolean isPrime = isPrime(i); // check i for prime

            if(isPrime) {
                primeCounter++;
                System.out.println("Prime number " + primeCounter + " is " + i);
            }

            // exit loop after max. number of primes
            if(primeCounter > maxNumberOfPrimesToFind) {
                System.out.println("Exited loop after max number of found primes!");
                break;
            }

        }
    }


    /**
     * isPrime()
     * Test if a number is a prime number
     * @param n     Number to be tested
     * @return      Boolean: true = prime; false != prime
     */
    public static boolean isPrime(int n) {

        if(n == 1) {
            return false;
        }

        //for(int i = 2; i <= n/2; i++) {   // correct but slower due to more loops required
        for(int i = 2; i <= (long) Math.sqrt(n); i++) {   // performance optimized due to less loops required
            if(n % i == 0) {
                return false;
            }
        }

        return true;
    }


    /**
     * calculateInterest()
     * Calculate the interest (= Zins)
     * @param amountOfMoney Amount of money to be calculated
     * @param interestRate Interest Rate in %
     * @return Amount of grown money
     */
    public static double calculateInterest(double amountOfMoney, double interestRate) { // interest rate = Zinssatz
        return (amountOfMoney * (interestRate / 100));
    }
}

```