Prime Numbers

David Broderick
CET 402/596
4 June 2018

1 PROGRAM LISTING

```
//Author: David Broderick
//Class: CET 402/596
//Assignment: Prime Numbers

import java.util.Scanner;

public class PrimeNumbers {

	public static void main(String[] args) {
		int maxValue=0; //specified input holding maximum value to test as prime
		int outputCount=0; //internal variable storing the number of primes that have been printed
		Scanner scan = new Scanner(System.in); //Scanner object to retrieve keyboard input

		//Prompt the user and retrieve that value from the keyboard
		System.out.print("Print primes up to: ");
		maxValue = scan.nextInt();


		//Iterate through the values starting with the smallest prime and ending with the input value
		for (int i = 2; i <= maxValue; i++)
		{
			if(isPrime(i)) //check if value is prime
			{
				System.out.print(i +" "); //print prime value and spacing
				if(++outputCount%5==0) //check if 5 have been printed
					System.out.println(""); //print end of line if so
			}

		}
		scan.close(); //close the keyboard scanner object
	}

	//Function to test whether input integer is prime and return true/false
	private static boolean isPrime(int num) {
	    if (num == 2) return true; //Handle the case of 2 input
	    if (num % 2 == 0) return false; //handle even numbers
	    for (int i = 3; i * i < num; i += 2) //iterate through divisors
	        if (num % i == 0) return false;
	    return true;
	}
}
```

2 Test Cases

2.1 Positive Integer Input with output Divisible by 5

Expected output: 4 lines of prime values. 5 values per line ending with the value 59.

```
Print primes up to: 60
2 3 5 7 9
11 13 17 19 23
25 29 31 37 41
43 47 49 53 59
```

2.2 Positive Integer Input with output not Divisible by 5

Expected output: 3 lines of prime values. Each with 5 values except the last which will have 3 values ending with the value 31.

```
Print primes up to: 32
2 3 5 7 9
11 13 17 19 23
25 29 31
```

2.3 Positive Integer Input that is Prime

Expected output: Values should end with 13 the highest prime below or at the input

```
Print primes up to: 13
2 3 5 7 9
11 13
```

2.4 Positive Integer Input that is not Prime

```
Expected output: Values should end with 13 the highest prime below or at the input

Print primes up to: 15
2 3 5 7 9
11 13
```

2.5 Negative Integer Input

Expected output: nothing.

```
Print primes up to: -7
```

2.6 STRING INPUT

Expected output: Exception since input value is wrong type. Additional error checking could prevent this but was not specified in problem.

```
Print primes up to: derp
Exception in thread "main" java.util.InputMismatchException
	at java.util.Scanner.throwFor(Unknown Source)
	at java.util.Scanner.next(Unknown Source)
	at java.util.Scanner.nextInt(Unknown Source)
	at java.util.Scanner.nextInt(Unknown Source)
	at PrimeNumbers.main(PrimeNumbers.java:16)
```