Reflection:

This is my first official java coding experience. Everything is going well so far. I think lab 1 is very useful in this assignment. The basic pattern of this assignment is very similar to the example available in lab 1. Hence, it does not cost me a lot of time to write the code. The only thing I need to think is to figure out the algorithm for isPrime(). Here, the knowledge in Discrete Math is very useful. I have learned in Discrete Math that if a number is not a prime number, it will have the factor that is smaller or equal to its square root.

At the same time, I think to make use the help of Emacs is also very important. At the first, I misplace the last “return True” in isPrime(). Emacs helps me figure out the error quickly.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Assignment 2

\* Name: Siyan Wang

\* Email: swang2@haverford.edu

\* Course: CS 206

\* Submitted: 9/15/2015

\*

\* The source code for printing all prime numbers between 1 and 100.

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Creat a class called Prime to check whether a number is a prime number and print all prime numbers between 1 and 100

\* author Siyan Wang(swang2@haverford.edu)

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class Primes{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Creat a function called isPrime to check whether a number is a prime number

\* author Siyan Wang(swang2@haverford.edu)

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public static boolean isPrime(int number){

// Returns true if n is a prime, false otherwise

if (number==1){

// 1 is not a prime

return false;

}

else{

// if n is not a prime, it will have a factor <= the square root of n

for (int j=2; j<=(int) Math.sqrt(number); j++){

// check whether j is the factor of n

if(number%j == 0){

return false;

}

}

}

// if n does not have a factor that is larger than 1 and smaller than sqrt(n), it is a prime

return true;

}//end of isPrime()

public static void main (String [] args){

// Print out all prime numbers between 1 and 100

for (int i=1; i<=100; i++){

if (isPrime(i)){

System.out.println(i);

}

}

}// end of main()

}// end of class Primes()

Output:

2

3

5

7

11

13

17

19

23

29

31

37

41

43

47

53

59

61

67

71

73

79

83

89

97