Tu Luan

Professor Copper

CS206

3:34:08 PM EDT

Sep 15, 2015

Reflection:

It is hard at the beginning since I used to try to build the test and print method together under the main class. I always lost myself inside such code when checking and could not fix the syntax bugs. But I find it is clearer to build a test method in another class and call this method in the main class. So finally I wrote them separately and successfully got the wanted result.

Also, I used to use ”elif” in the code, but I found java cannot recognize it. So I looked up the right “else if” syntax online and got the correct one.

The number list is:

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

Code:

//Assignment 02

//Name: Tu Luan

//E-mail: tluan@brynmawr.edu

//Course: Cs206 Professor Cooper's section

//Submitted: 09/15/2015

//Goal:build isPrime class to find out all prime number in 1-100

public class isPrime{

public static void main(String[]args){

//since we all know 1 is not regarded as a prime number, I start from

//2 here.

//since 100 is contained, I used 101 as its end point.

for(int a=2;a<101;a++){

if (a==2){

System.out.println(a);

}

//2 is an exception here so I print it seperately

//print out all the numbers when they are tested as prime numbers

//by using the testPrime method defined below.

else if(testPrime(a)!=false){

System.out.println(a);

}

}

}

//set a testPrime class to test if the number is prime

public static boolean testPrime(int n){

//start from 2(included) to itself(excluded), test if it can be factored

//into other numbers expect 1 and itself. If yes, then it is not a prime // number, then return false

for(int i=2;i<n;i++){

if( n%i==0){

return false;

}

}

//after the factor test, if it still not return false, then return true

//to consider it as a prime number

return true;

}

}