Weslyn Wagner EE 5343 UTSA: Java Assignment #1 9/11/2016

1. a) The Java platform is independent because all of the platform dependent code runs on a Java Virtual Machine (JVM). Since the JVM is standard and in itself can run on any platform, the Java language is independent of the underlying architecture and only dependent on the JVM, which separates the two.

b) Using command line arguments is taking user input at runtime and then using that input in the program. The command line arguments also allow users to call the program. At runtime, arguments for the main method can be entered when the program is called or be entered after the program is called if the appropriate java utils are utilized.

c) Strings being immutable means you cannot change them, or mutate them. Unlike arrays and lists, in which you can change the contents, strings do not allow you to change their contents of chars. You can create new strings or convert the string to another data type, but you cannot explicitly alter a string, making it immutable.

2.

b

a

Get the inputs from user, and sort

(a must be bigger than b)

Divide a by b to get remainder, r

If r!=0, then divide b by r

If r==0, then GCD is b

/\*

\* September 10, 2016

\*Version 1.0

\*EE 5343 UTSA: Java Assignment #1 Problem 2

\*Student: Weslyn Wagner

\*/

import java.util.Scanner;

class EuclideanGCD{

public static void main(String[] args){

Scanner input = new Scanner(System.in);

System.out.print("Enter the first number: ");

int a = input.nextInt();

System.out.print("Enter the second number: ");

int b = input.nextInt();

int[] numbers = new int[]{a,b};

if(a<b){

numbers[0]= b;

numbers[1]= a;

}

GCD(numbers[0],numbers[1]);

}

public static int GCD(int a, int b){

int r;

r = a%b;

System.out.println("r is :"+r);

if(r==0){

System.out.println("The GCD is :"+ b);

}

else{

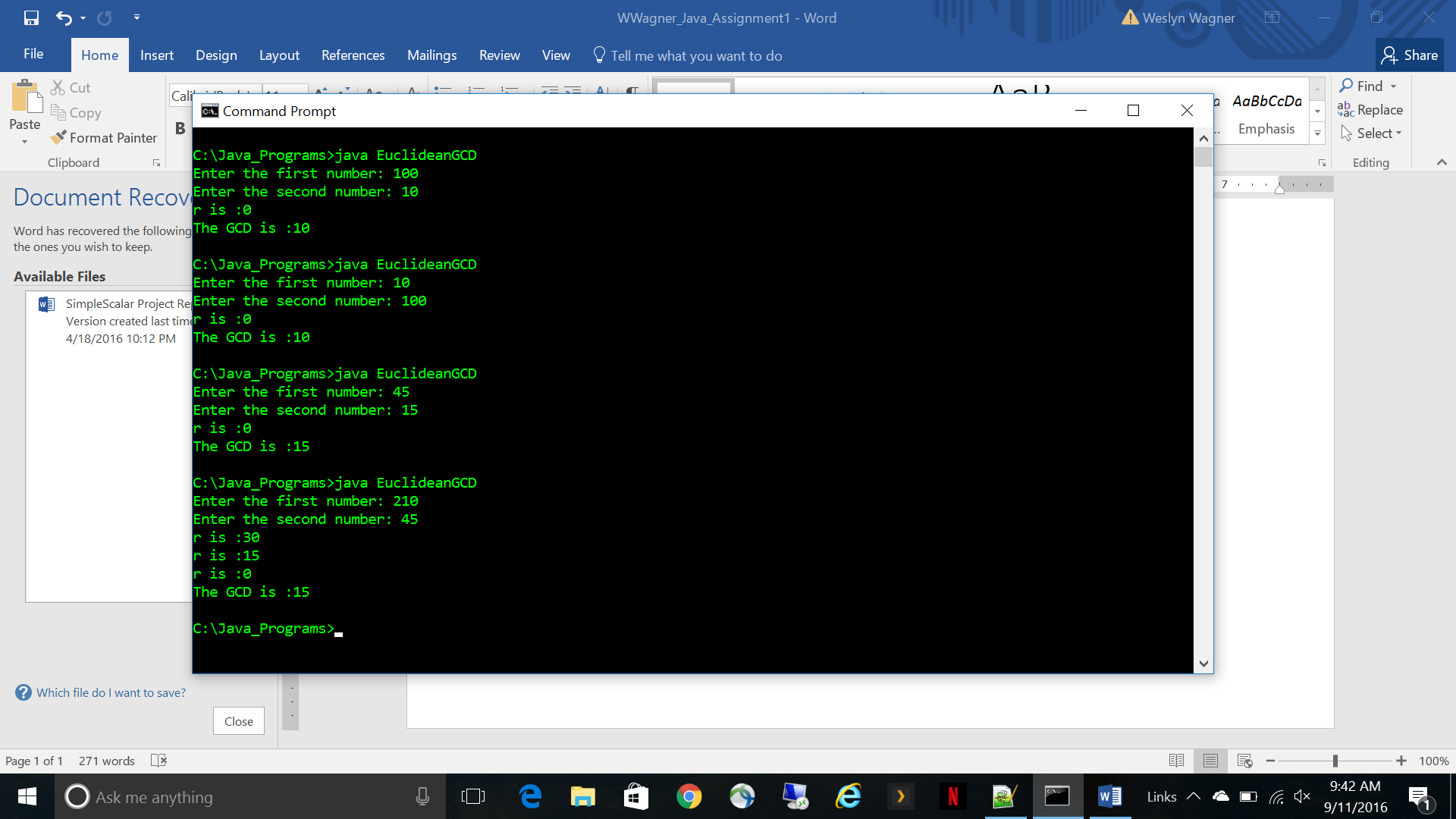
GCD(b,r);

}

return b;

}

}



3.

Desired accuracy

Get the desired accuracy from user as a decimal input

Use estimation formula to find value of pi

If diff between pi and estimation of pi < desired accuracy, add iterative term to estimation

If diff between pi and estimation of pi > desired accuracy, return estimation

If even iterative term, subtract (4/j) from estimation, where j is iterative denominator term

If odd iterative term, add (4/j) to estimation, where j is iterative denominator term

/\*

\* September 10, 2016

\*Version 1.0

\*EE 5343 UTSA: Java Assignment #1 Problem 3

\*Student: Weslyn Wagner

\*/

import java.util.Scanner;

import java.math.\*;

class estimatePi{

public static void main(String[] args){

Scanner input = new Scanner(System.in);

System.out.print("Enter the desired accuracy: ");

double acc = input.nextDouble();

System.out.println("Estimate is: "+ estimate(acc));

}

public static double estimate(double acc){

double numerator =4.0;

double pi;

double pi2;

pi = 3.1415926535899793238462;

pi2= numerator;

double diff = Math.abs(pi-pi2);

double iter = 2.0;

while(diff>=acc){

double j = 2\*iter-1;

if(iter%2==0){

pi2= pi2-(numerator/j);

}

else{

pi2= pi2+(numerator/j);

}

iter+=1.0;

diff = Math.abs(pi-pi2);

//System.out.println("Pi is: "+ pi);

System.out.println("Pi2 is: "+ pi2);

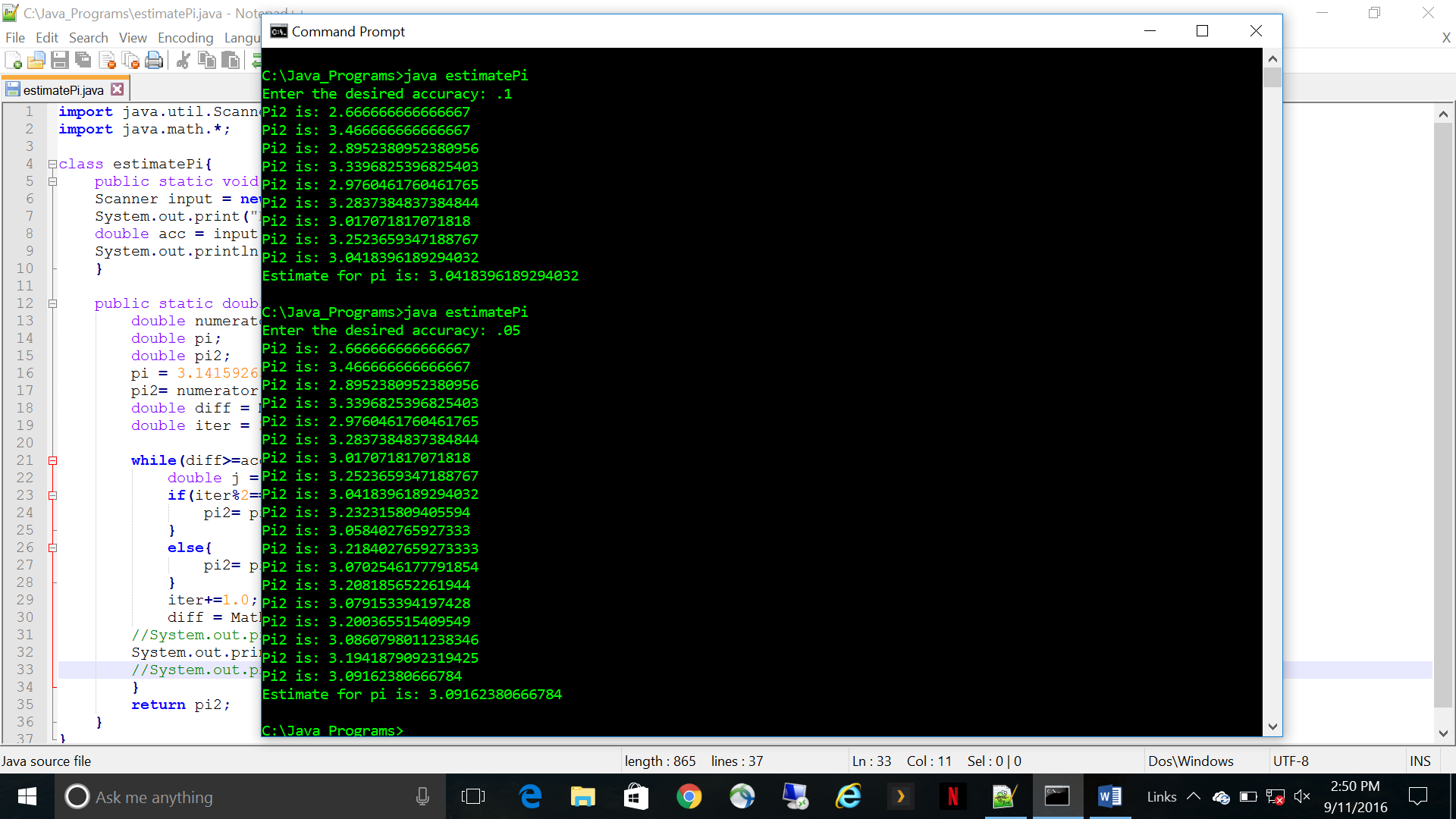
//System.out.println("The diff is: "+ diff);

}

return pi2;

}

}



4.

Take the initial amount of FB users and multiply by the first month’s growth rate (1.02)

Multiply the result by the growth rate after the first month (1.03); track how many times that number is compounded

If the result is > 2.5B, DONE. Print how many times compounded

If the result is < 2.5B

/\*

\* September 10, 2016

\*Version 1.0

\*EE 5343 UTSA: Java Assignment #1 Problem 4

\*Student: Weslyn Wagner

\*/

import java.math.\*;

class facebook{

public static void main(String[] args){

//Have to convert # users to BigDecimal to be able to multiply by a double

BigDecimal initUsers = new BigDecimal("1000000000");

//The growth rate for the first month

double rate = 1.02;

BigDecimal afterMonthOne = initUsers.multiply(BigDecimal.valueOf(rate));

System.out.println(afterMonthOne);

BigDecimal target = new BigDecimal("2500000000");

double new\_rate = 1.03;

int monthsAfter = 1;

//Keep multiplying the number of users by growth rate until the total is greater than 2.5B

//Use compareTo method to compare BigDecimals

while(target.compareTo(afterMonthOne)>0){

afterMonthOne = afterMonthOne.multiply(BigDecimal.valueOf(new\_rate));

System.out.println(afterMonthOne);

monthsAfter+=1;

}

System.out.println("Facebook will reach 2.5B users "+monthsAfter+" months after Oct. 2012");

}

}

