CSA0976 Java Programming

Name: C.Manjunath reddy

Reg No: 192011373

Assignment 3

1.Code:

import java.awt.\*; import java.util.\*; import javax.swing.\*;

public class ColorfulText extends JPanel implements Runnable { private static final long serialVersionUID = 1L; private int x, y; private String message; private Color color;

private Random random;

public ColorfulText() {

x = 50; y = 50;

message = "Hello, world!"; color = Color.BLACK; random = new Random();

}

@Override protected void paintComponent(Graphics g) { super.paintComponent(g);

g.setFont(new Font("Arial", Font.BOLD, 36));

g.setColor(color);

g.drawString(message, x, y);

}

@Override public void run() { while (true) { try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace(); }

color = new Color(random.nextInt(256), random.nextInt(256),

random.nextInt(256));

repaint();

} }

public static void main(String[] args) { JFrame frame = new JFrame("Colorful Text"); frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); frame.setSize(400, 200);

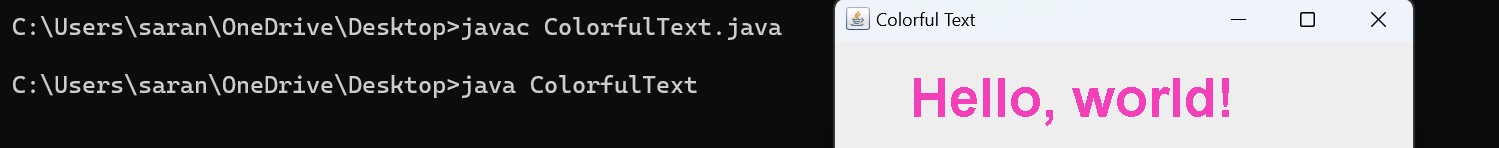
ColorfulText colorfulText = new ColorfulText(); frame.add(colorfulText); frame.setVisible(true);

Thread thread = new Thread(colorfulText);

thread.start();

}

} Output:



2.Code:

class Table

{ void printTable(int n)

{

synchronized(this)

{

for(int i=1;i<=5;i++)

{

System.out.println(n+"\*"+i+"="+(n\*i));

try

{

Thread.sleep(500);

} catch(Exception e)

{

System.out.println(e);

}

}

}

}

} class Mythread1 extends Thread

{

Table t;

Mythread1(Table t)

{

this.t=t;

} public void run()

{

t.printTable(5);

} } class Mythread2 extends Thread

{

Table t;

Mythread2(Table t)

{

this.t=t;

} public void run()

{

t.printTable(10);

} } class Use {

public static void main(String arg[])

{

Table obj=new Table();

Mythread1 th1=new Mythread1(obj);

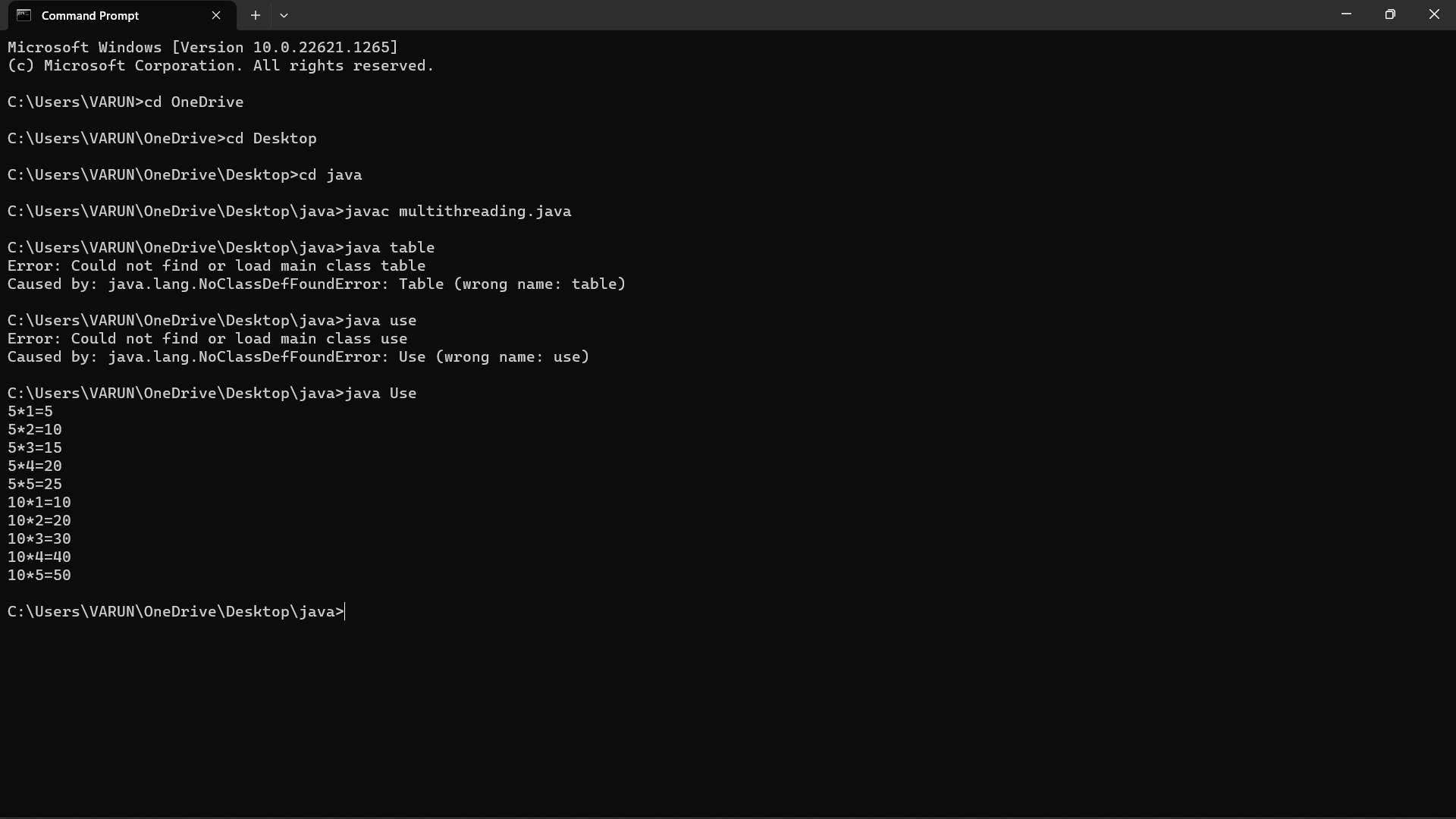
Mythread2 th2=new Mythread2(obj);

th1.start();

th2.start();

}

} Output:



3.Code:

import java.io.\*; import java.util.\*; class ugly

{ public static boolean ugl(int n)

{

if(n<=0)

{

return false; } while(n%2==0)

{

n/=2; } while(n%3==0)

{

n/=3; } while(n%5==0)

{

n/=5; }

return n==1; } public static void main(String arg[])

{

int n;

Scanner a=new Scanner(System.in); System.out.print("Enter a numnber :");

n=a.nextInt(); if(ugl(n))

{

System.out.print("True the given number is a ugly number");

}

else

{

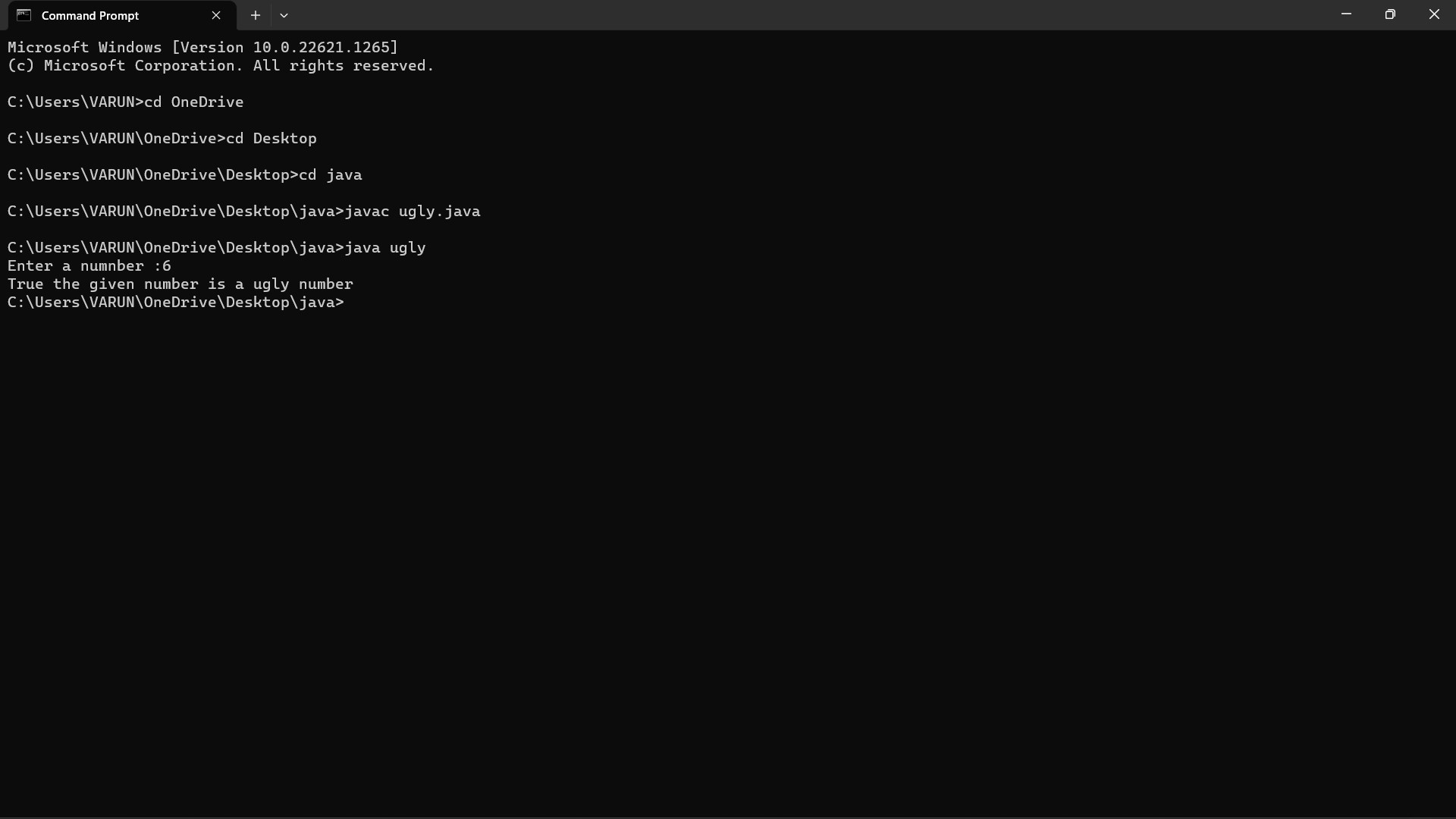
System.out.print("False the given number is not a ugly

number");

}

}

} Output:



4.Code:

import java.io.\*; import java.util.\*; class fiboseries

{ public static void main(String arg[])

{

int n;

Scanner a=new Scanner(System.in); System.out.print("Enter a number :");

n=a.nextInt(); if(n<0)

{

System.out.println("Enter a positive Integer ");

}

else

{

System.out.print("Output :"+fibonacci(n));

} }

public static int fibonacci(int n)

{

if(n==1||n==0)

{

return(n);

}

else

{

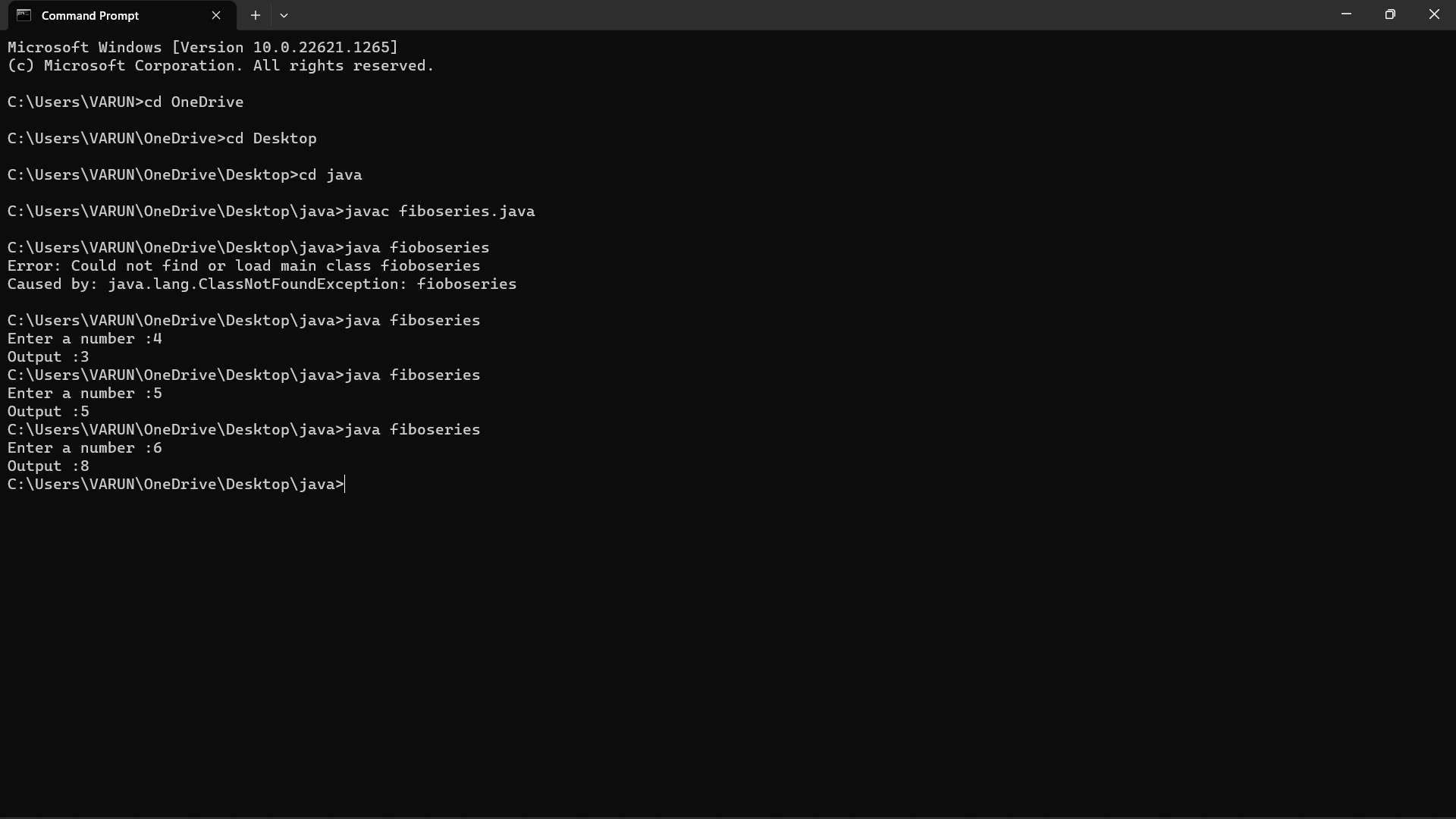
return(fibonacci(n-1)+fibonacci(n-2));

}

}

}

Output:



5.Code:

class duplicate

{

// Function to remove duplicate elements // This function returns new size of modified // array. static int removeDuplicates(int arr[], int n)

{

// Return, if array is empty // or contains a single element if (n==0 || n==1) return n;

int[] temp = new int[n];

// Start traversing elements

int j = 0;

for (int i=0; i<n-1; i++)

// If current element is not equal

// to next element then store that // current element if (arr[i] != arr[i+1])

temp[j++] = arr[i];

// Store the last element as whether

// it is unique or repeated, it hasn't // stored previously temp[j++] = arr[n-1];

// Modify original array for (int i=0; i<j; i++) arr[i] = temp[i];

return j; }

public static void main (String[] args)

{ int arr[] = {10, 20, 20, 30, 40, 40, 40, 50, 50}; int n = arr.length;

n = removeDuplicates(arr, n); // Print updated array

for (int i=0; i<n; i++)

System.out.print(arr[i]+" ");

}

}

Output:

