**Assignment.**

1.break

public class 11{

public static void main(String[] args) {

//outer loop

for(int i=1;i<=3;i++){

//inner loop

for(int j=1;j<=3;j++){

if(i==2&&j==2){

break;

}

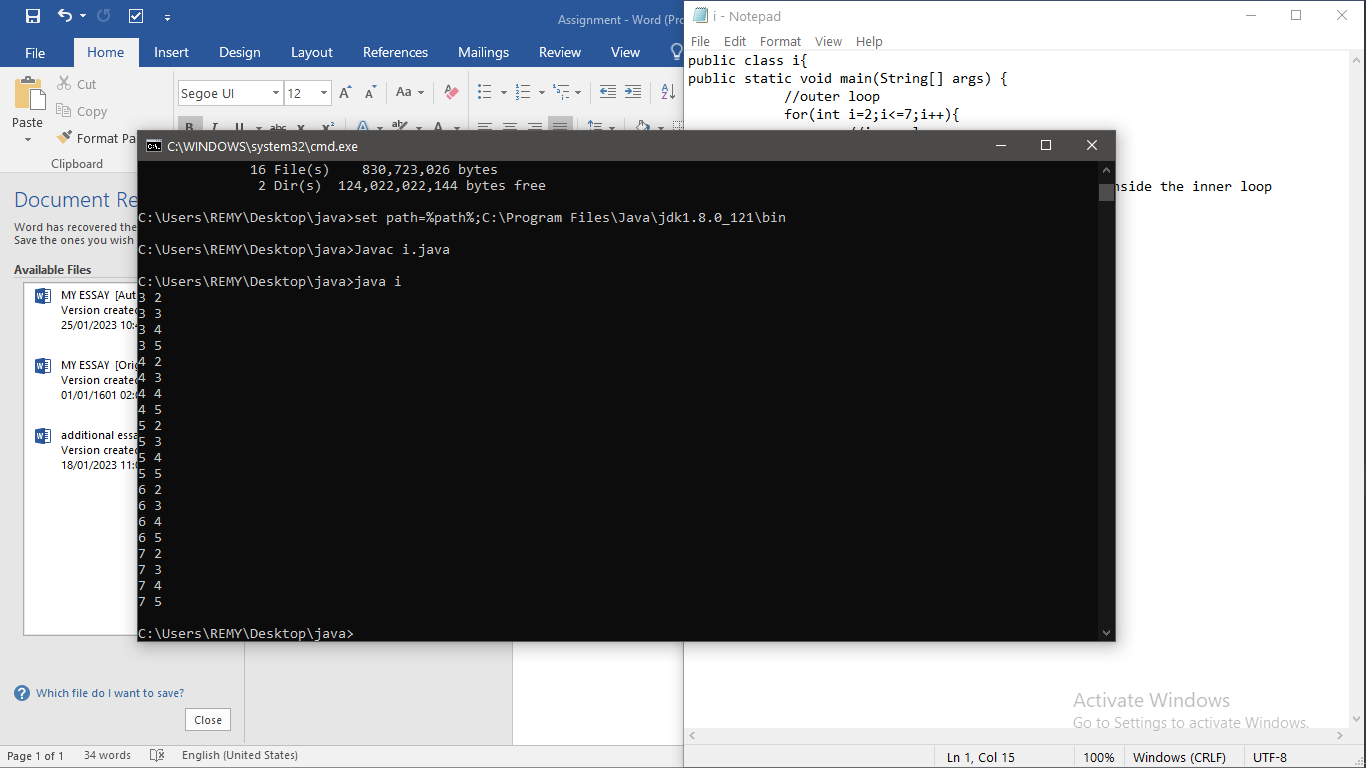
System.out.println(i+" "+j);

}

}

}

}



2.pyramid

public class x {

public static void main(String[] args) {

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

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

System.out.print("# ");

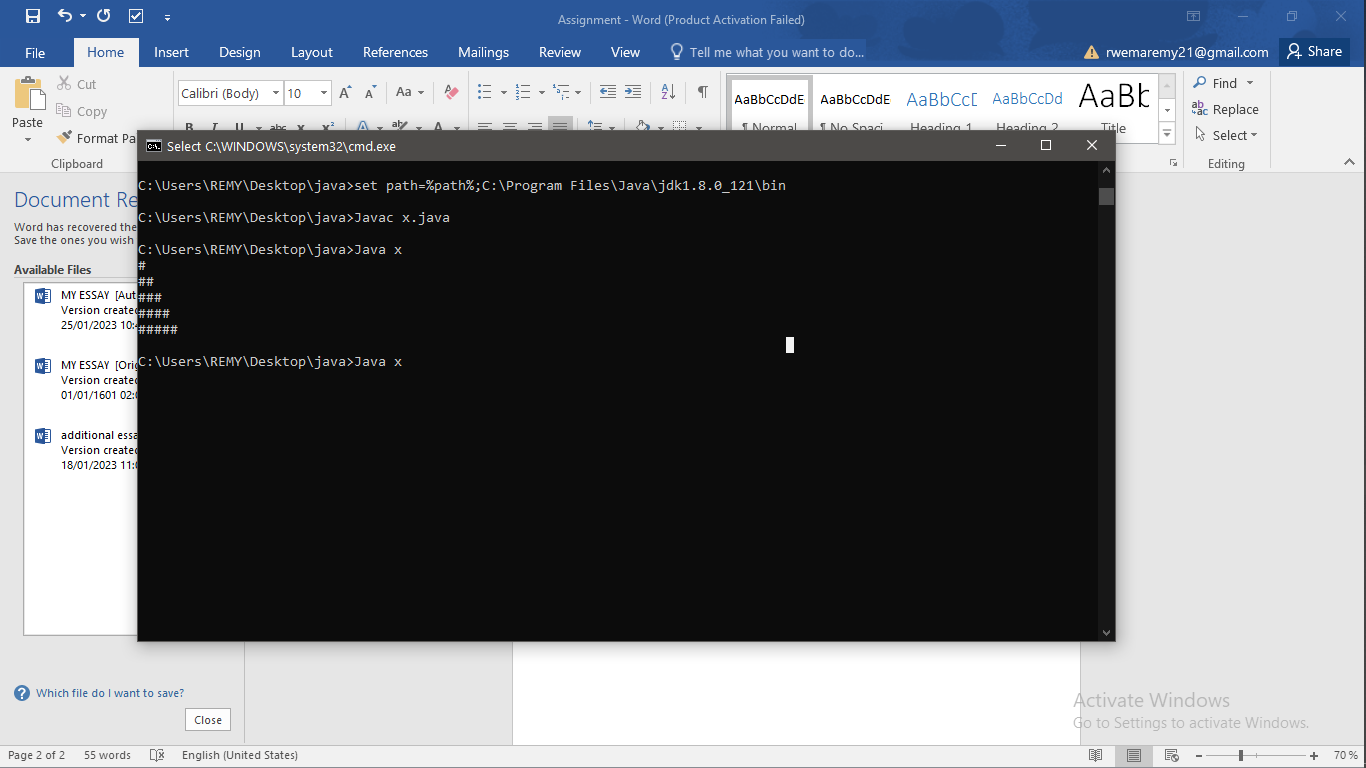
}

System.out.println();

}

}

}



3.if

public class q {

public static void main(String[] args) {

int age=16;

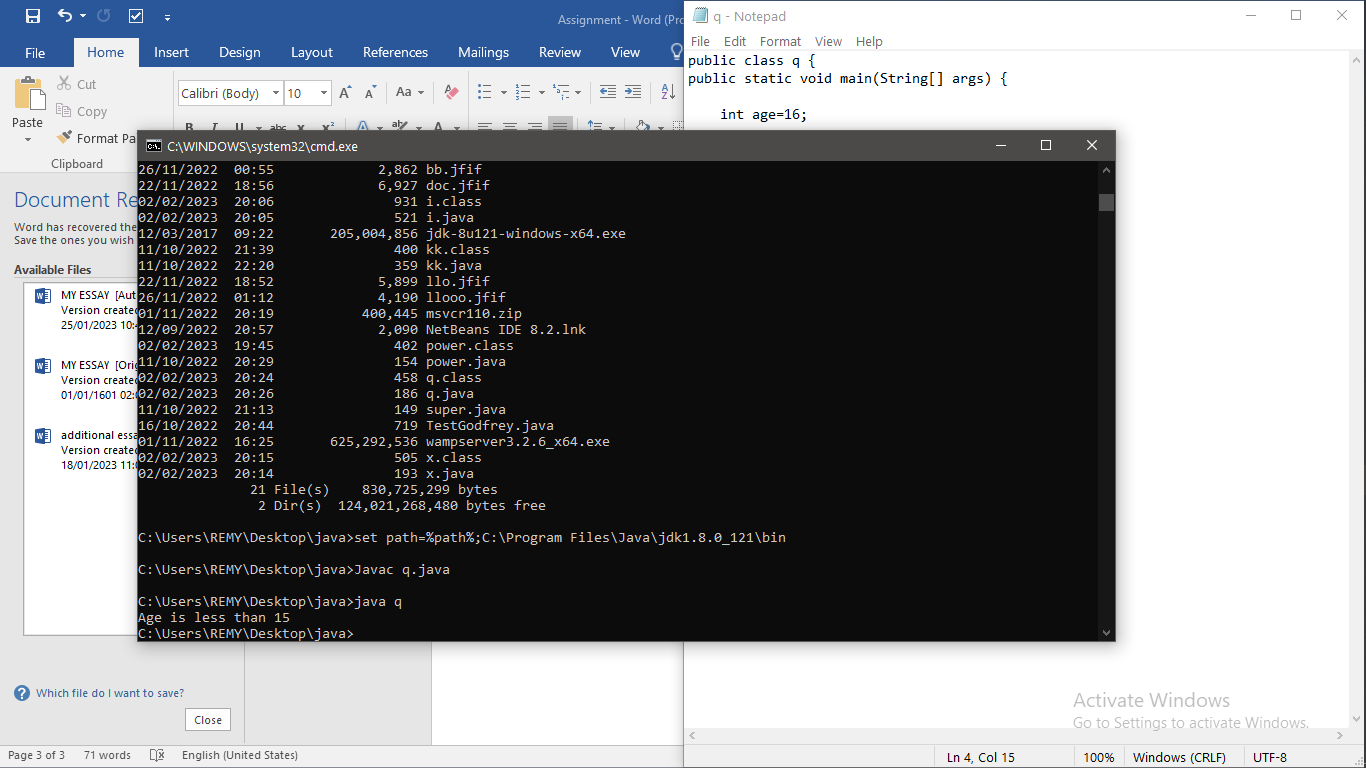
if(age>22){

System.out.print("Age is greater than 15");

}

}

}



4.clonning

class we implements Cloneable{

int rollno;

String name;

we(int rollno,String name){

this.rollno=rollno;

this.name=name;

}

public Object clone()throws CloneNotSupportedException{

return super.clone();

}

public static void main(String args[]){

try{

we s1=new we(22,"bruce");

we s2=(we)s1.clone();

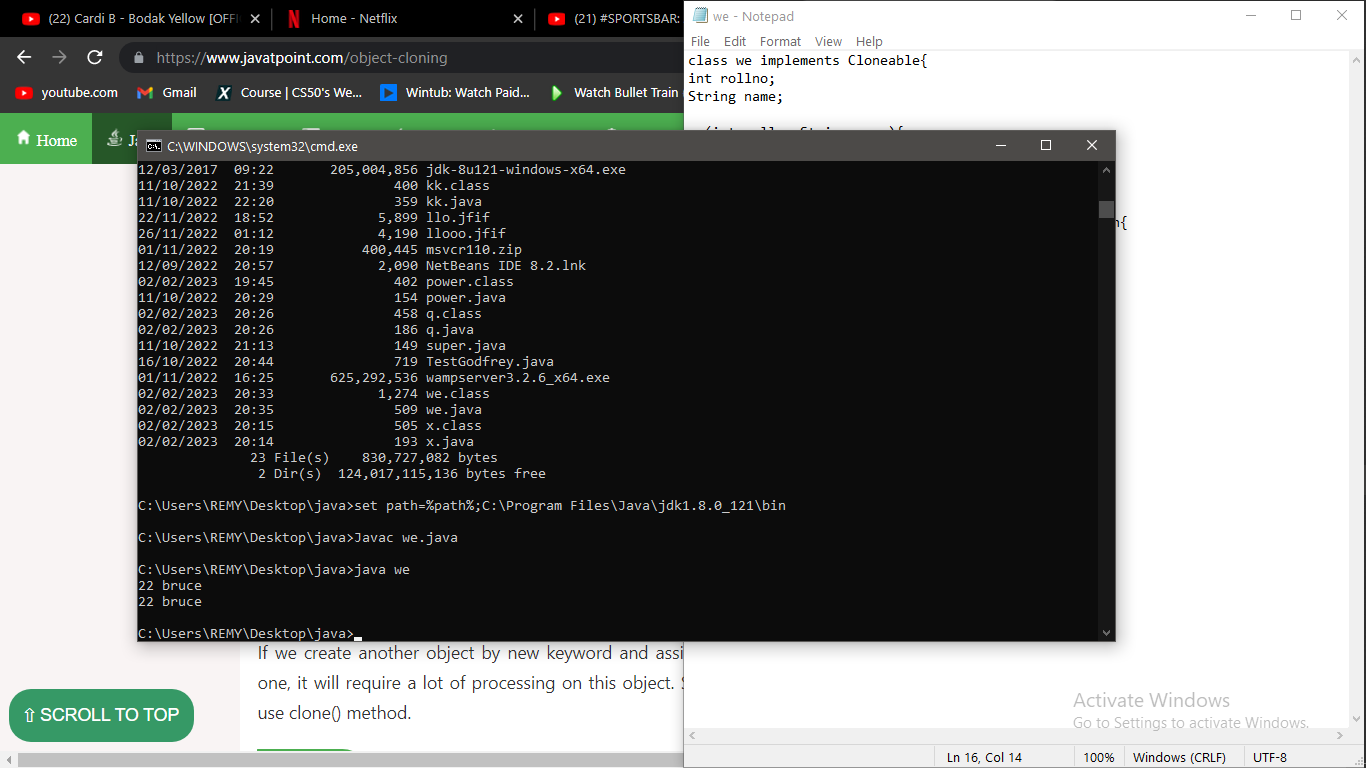
System.out.println(s1.rollno+" "+s1.name);

System.out.println(s2.rollno+" "+s2.name);

}catch(CloneNotSupportedException c){}

}

}



5.call by value

class re{

int data=76;

void change(re r){

r.data=r.data+24;

}

public static void main(String args[]){

re r=new re();

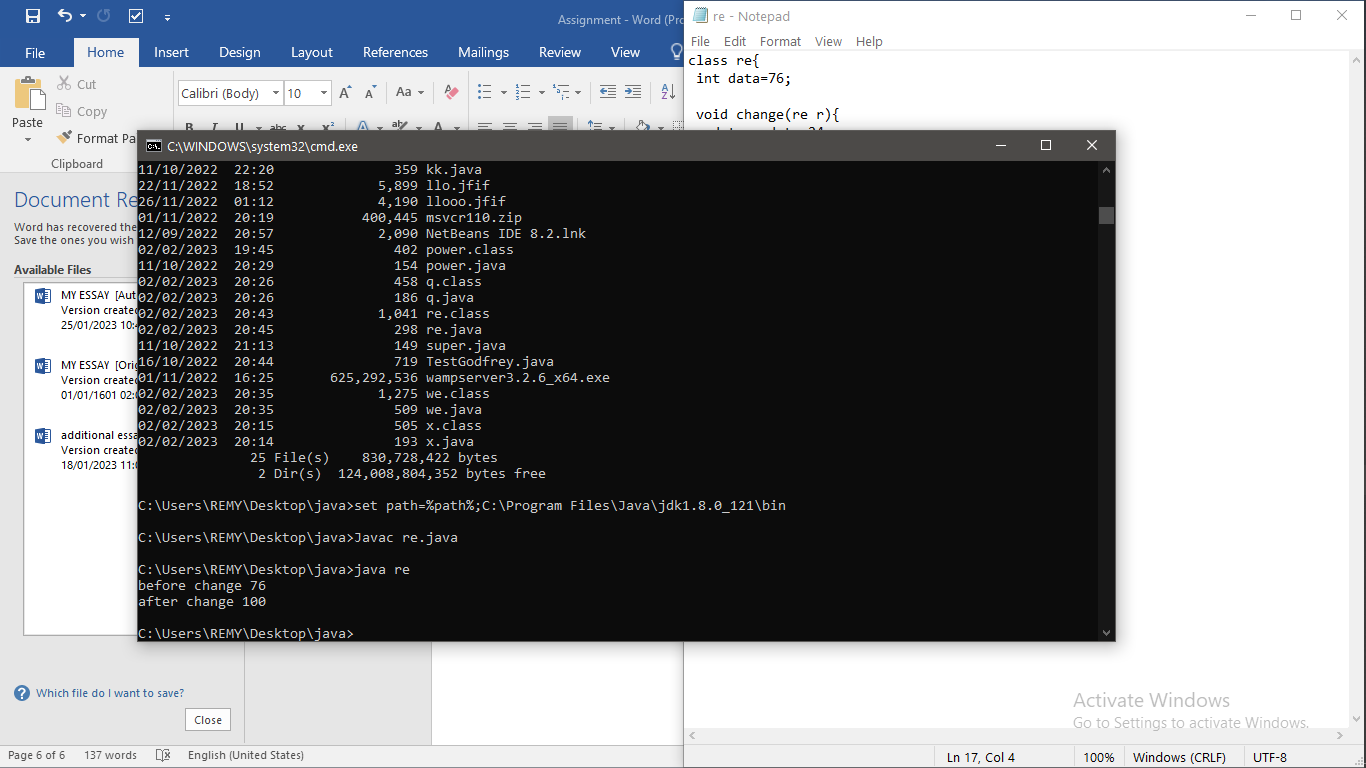
System.out.println("before change "+r.data);

r.change(r);

System.out.println("after change "+r.data);

}

}



6.wrapp

public class wr{

public static void main(String args[]){

byte b=10;

short s=20;

int i=30;

long l=40;

float f=50.0F;

double d=60.0D;

char c='a';

boolean b2=true;

Byte byteobj=b;

Short shortobj=s;

Integer intobj=i;

Long longobj=l;

Float floatobj=f;

Double doubleobj=d;

Character charobj=c;

Boolean boolobj=b2;

System.out.println("---Printing object values---");

System.out.println("Byte object: "+byteobj);

System.out.println("Short object: "+shortobj);

System.out.println("Integer object: "+intobj);

System.out.println("Long object: "+longobj);

System.out.println("Float object: "+floatobj);

System.out.println("Double object: "+doubleobj);

System.out.println("Character object: "+charobj);

System.out.println("Boolean object: "+boolobj);

byte bytevalue=byteobj;

short shortvalue=shortobj;

int intvalue=intobj;

long longvalue=longobj;

float floatvalue=floatobj;

double doublevalue=doubleobj;

char charvalue=charobj;

boolean boolvalue=boolobj;

System.out.println("---Printing primitive values---");

System.out.println("byte value: "+bytevalue);

System.out.println("short value: "+shortvalue);

System.out.println("int value: "+intvalue);

System.out.println("long value: "+longvalue);

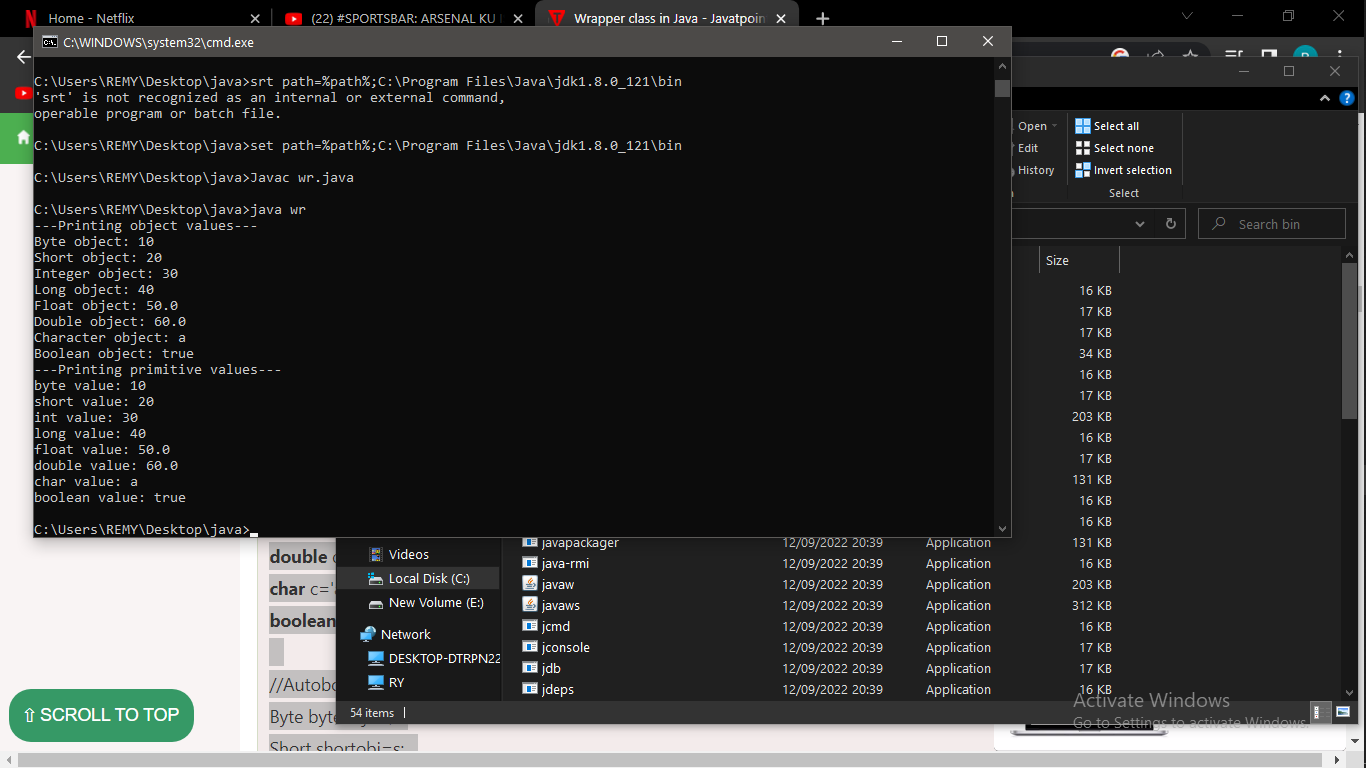
System.out.println("float value: "+floatvalue);

System.out.println("double value: "+doublevalue);

System.out.println("char value: "+charvalue);

System.out.println("boolean value: "+boolvalue);

}}



7.obstract class

abstract class s{

abstract void draw();

}

class Rectangle extends s{

void draw(){System.out.println("drawing rectangle");}

}

class Circle1 extends s {

void draw(){System.out.println("drawing circle");}

}

class sh{

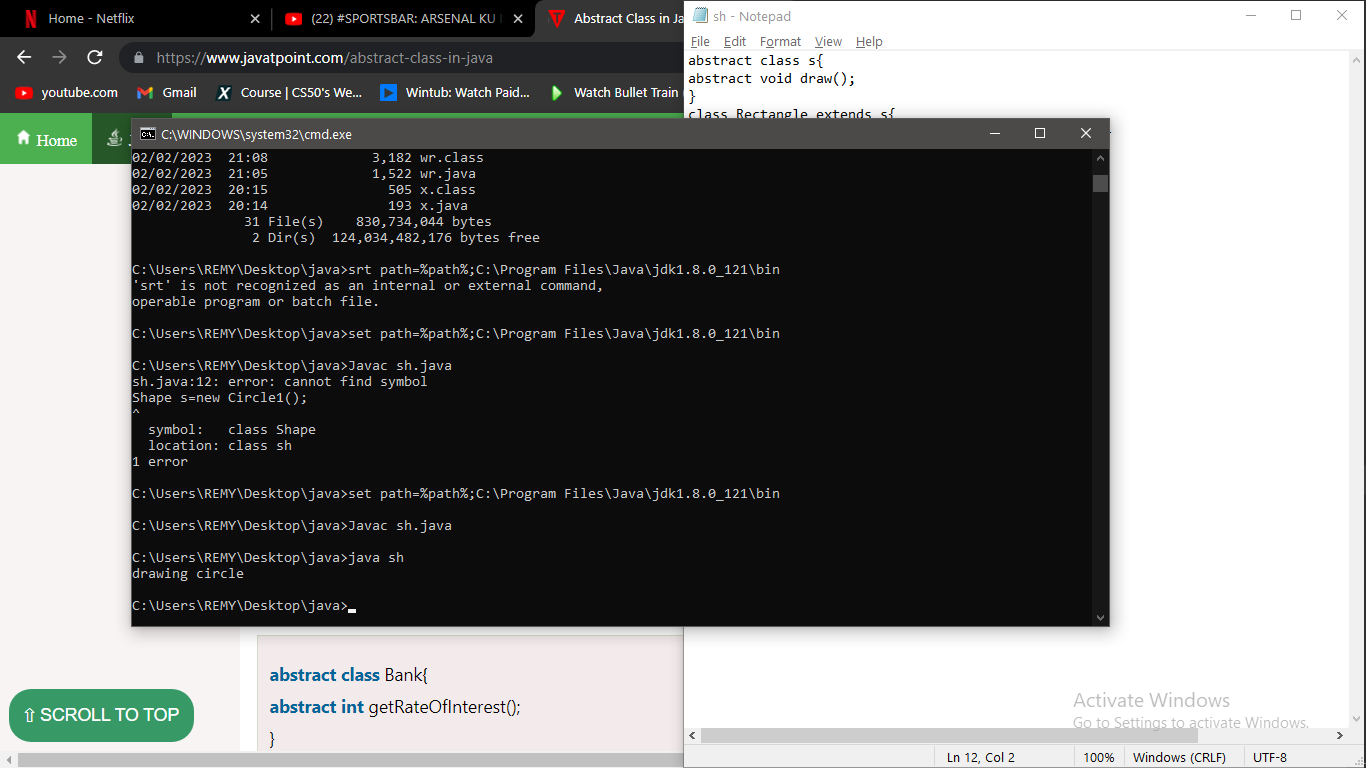
public static void main(String args[]){

s s=new Circle1();

s.draw();

}

}

8.interface.

interface

interface printable{

void print();

}

class A6 implements printable{

public void print(){System.out.println("Hello GUYS");}

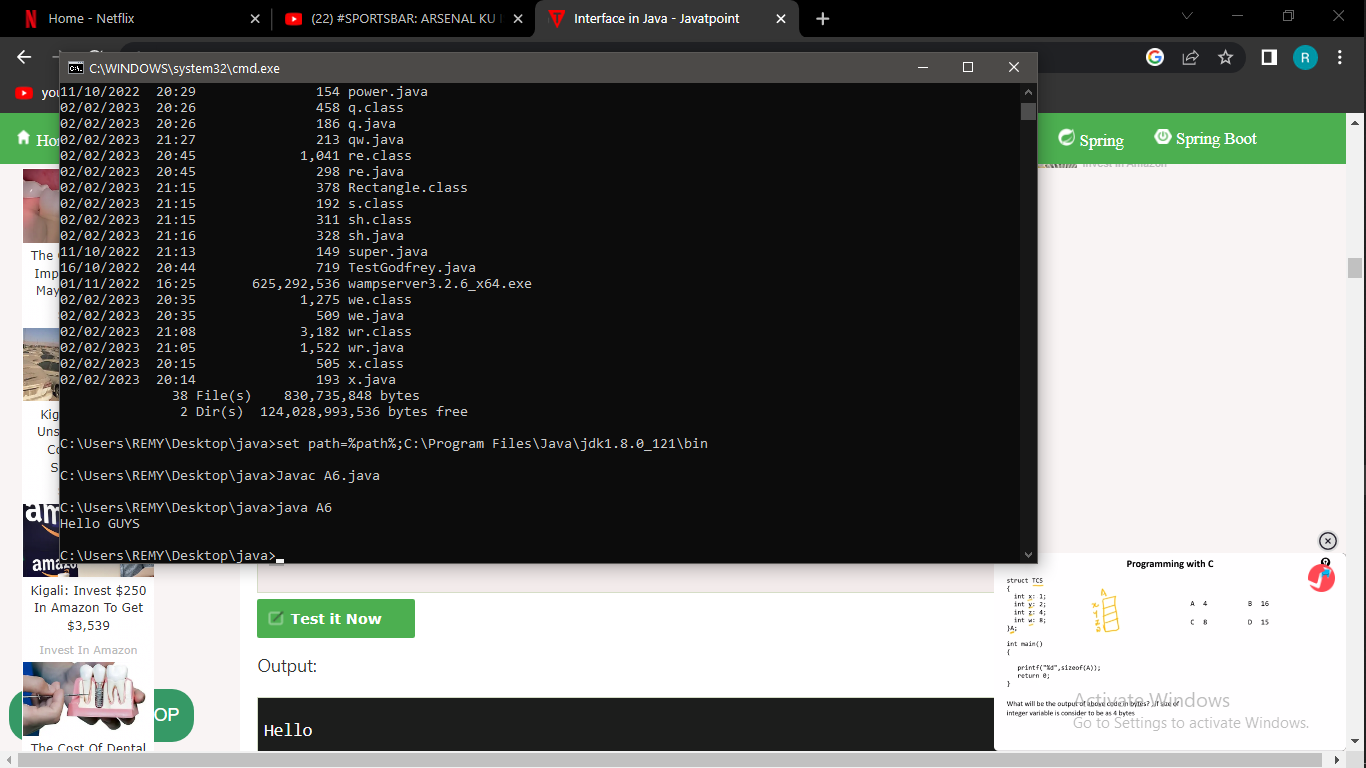
public static void main(String args[]){

A6 obj = new A6();

obj.print();

}

}



9.string builderand stringbuffer

public class opp{

public static void main(String[] args){

long startTime = System.currentTimeMillis();

StringBuffer sb = new StringBuffer("Java");

for (int i=0; i<10000; i++){

sb.append("Tpoint");

}

System.out.println("Time taken by StringBuffer: " + (System.currentTimeMillis() - startTime) + "ms");

startTime = System.currentTimeMillis();

StringBuilder sb2 = new StringBuilder("Java");

for (int i=0; i<10000; i++){

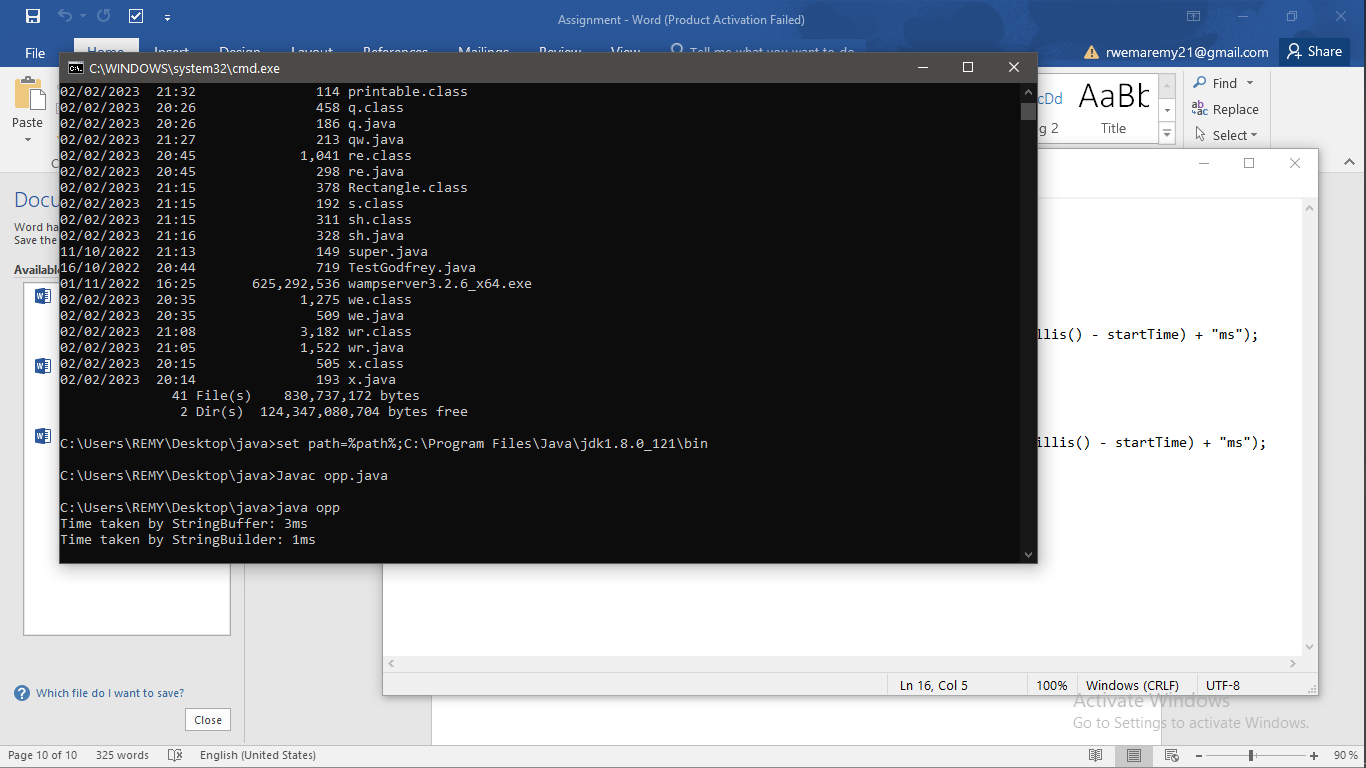
sb2.append("Tpoint");

}

System.out.println("Time taken by StringBuilder: " + (System.currentTimeMillis() - startTime) + "ms");

}

}



10.string replace

public class za{

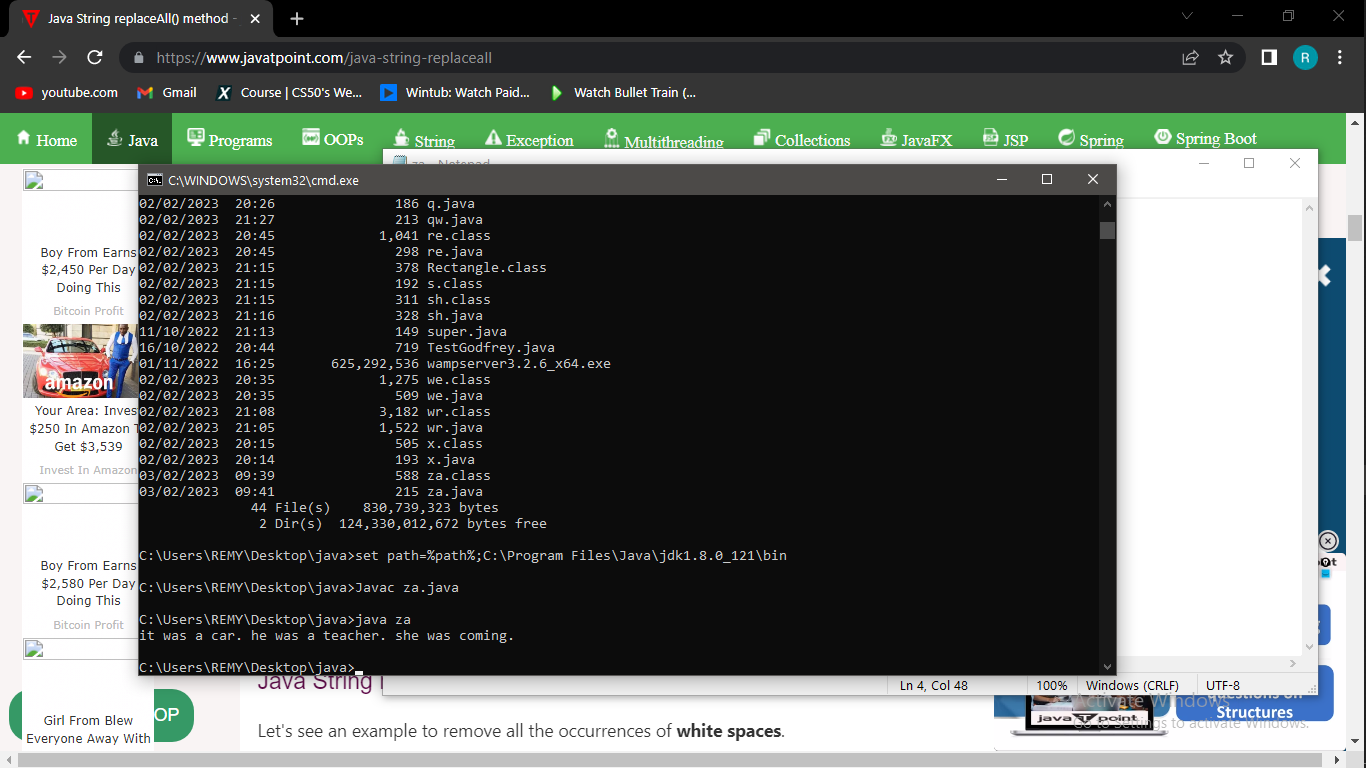
public static void main(String args[]){

String s1="it is a car. he is a teacher. she is coming.";

String replaceString=s1.replaceAll("is","was");

System.out.println(replaceString);

}}



11.split

public class vee{

public static void main(String args[]){

String s1="we have to learn coding to our most";

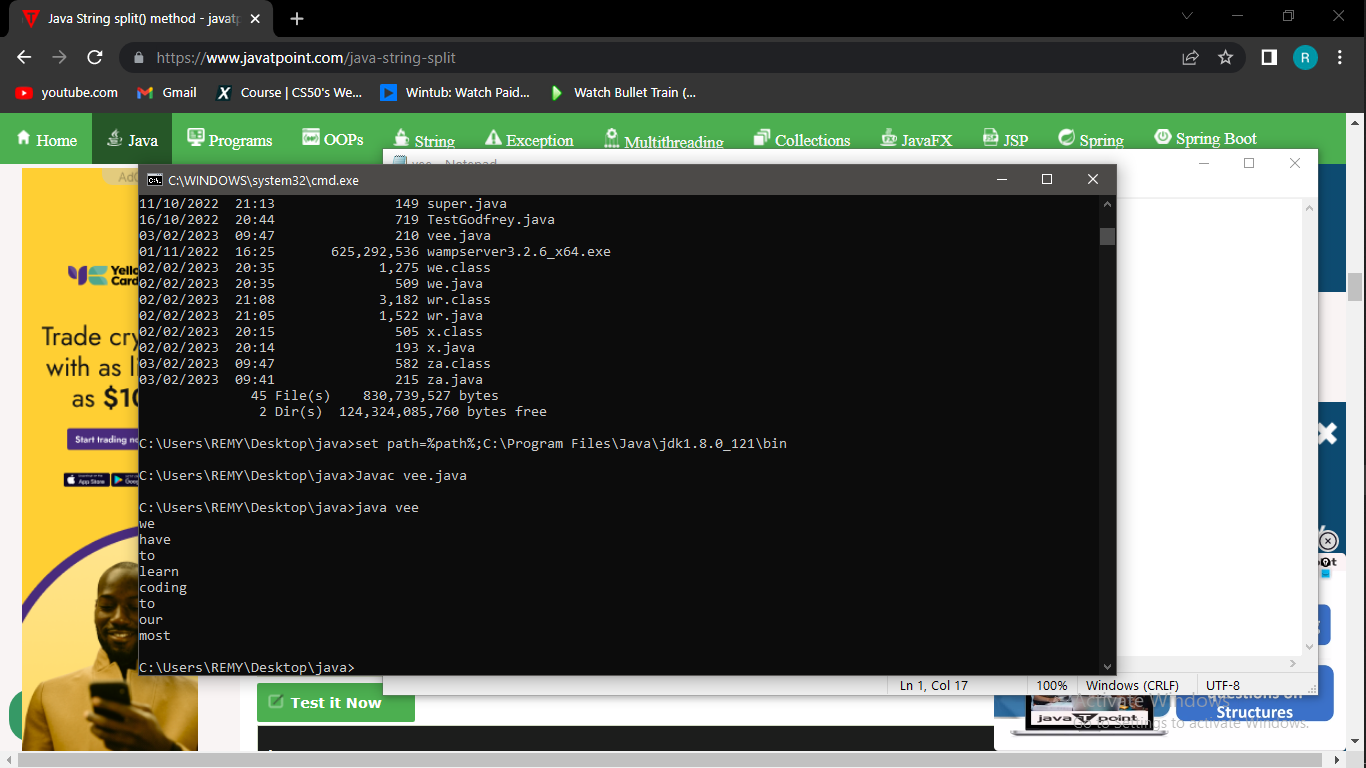
String[] words=s1.split("\\s");

for(String w:words){

System.out.println(w);

}

}}



12.nested class

public class ne{

static int data=90;

static class Inner{

static void msg(){System.out.println("data is "+data);}

}

public static void main(String args[]){

ne.Inner.msg();

}

}

