Names:KAYITESI ERNESTINE

REG:221020409

YEAR2

BIT

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;

}

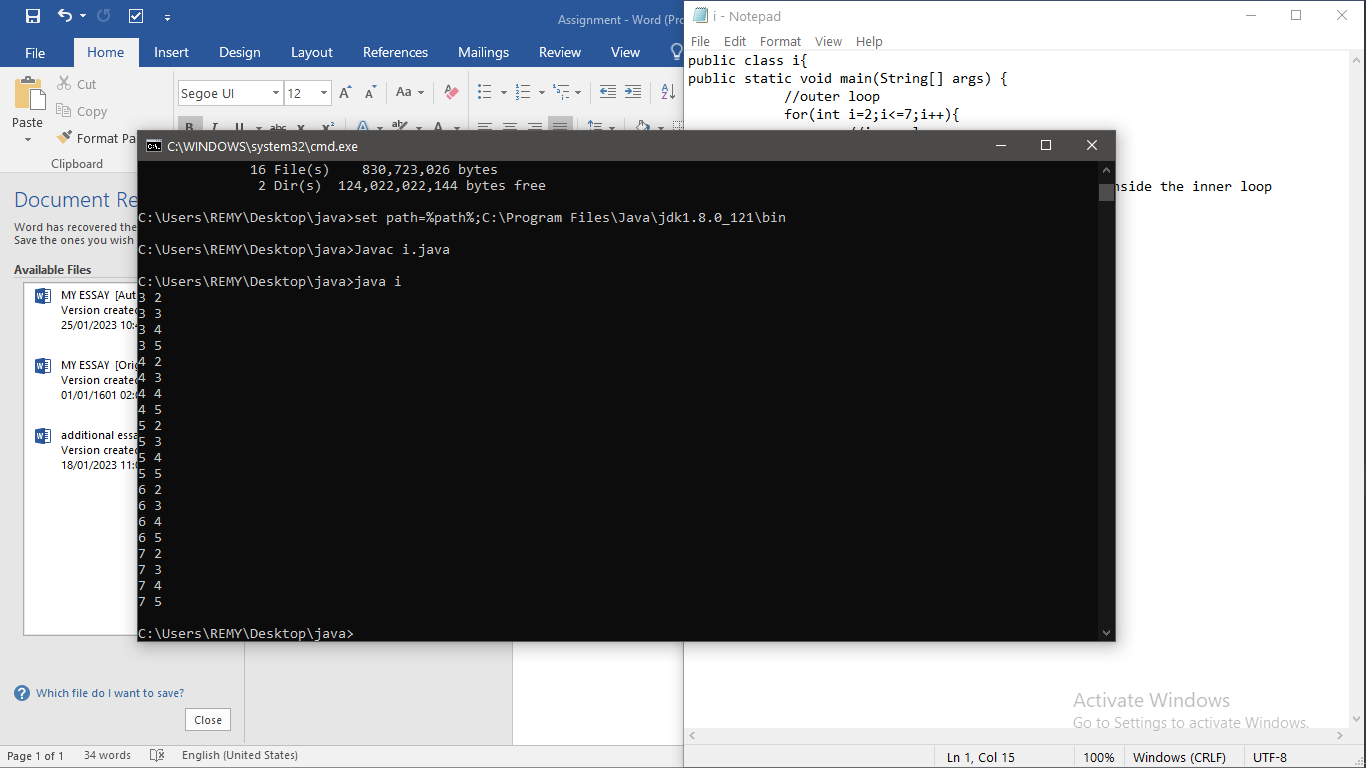
System.out.println(i+" "+j);

}

}

}

}



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("# ");

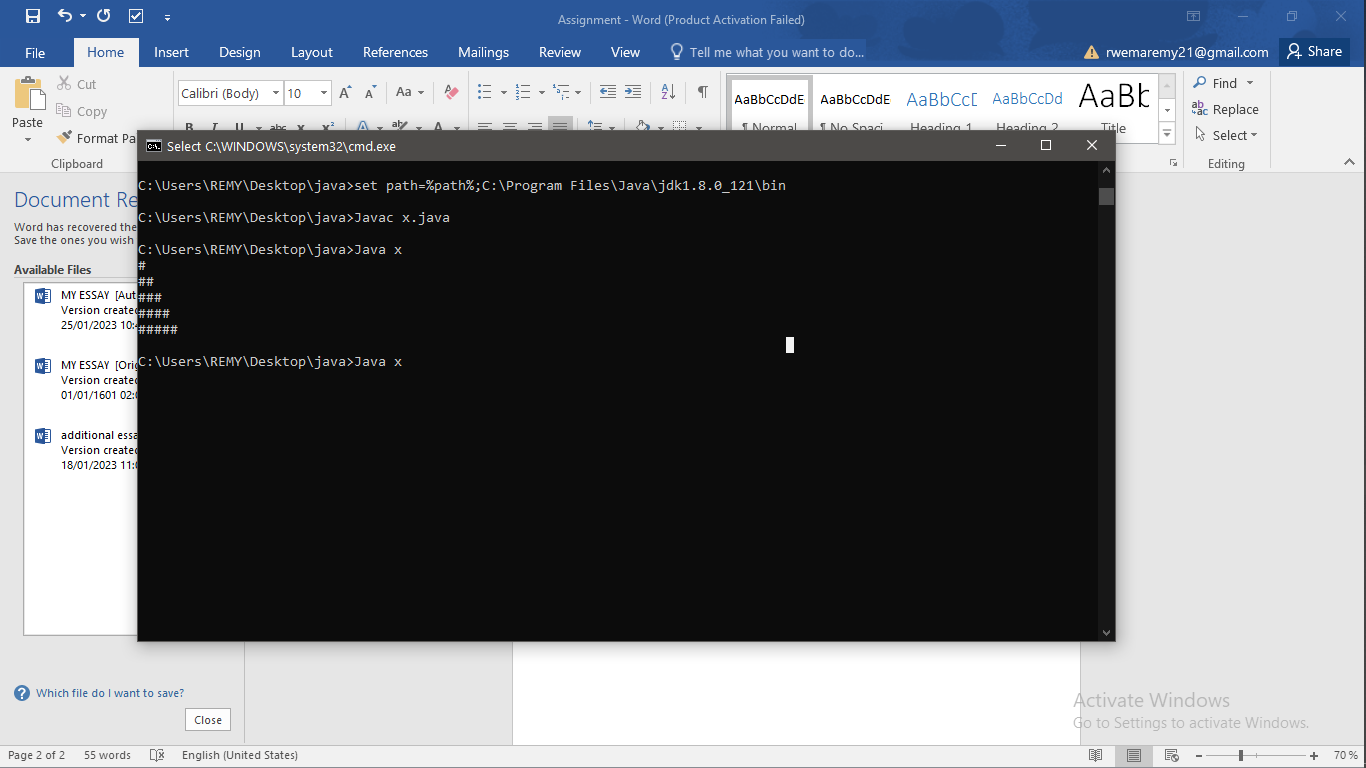
}

System.out.println();

}

}

}



3.if

public class q {

public static void main(String[] args) {

int age=16;

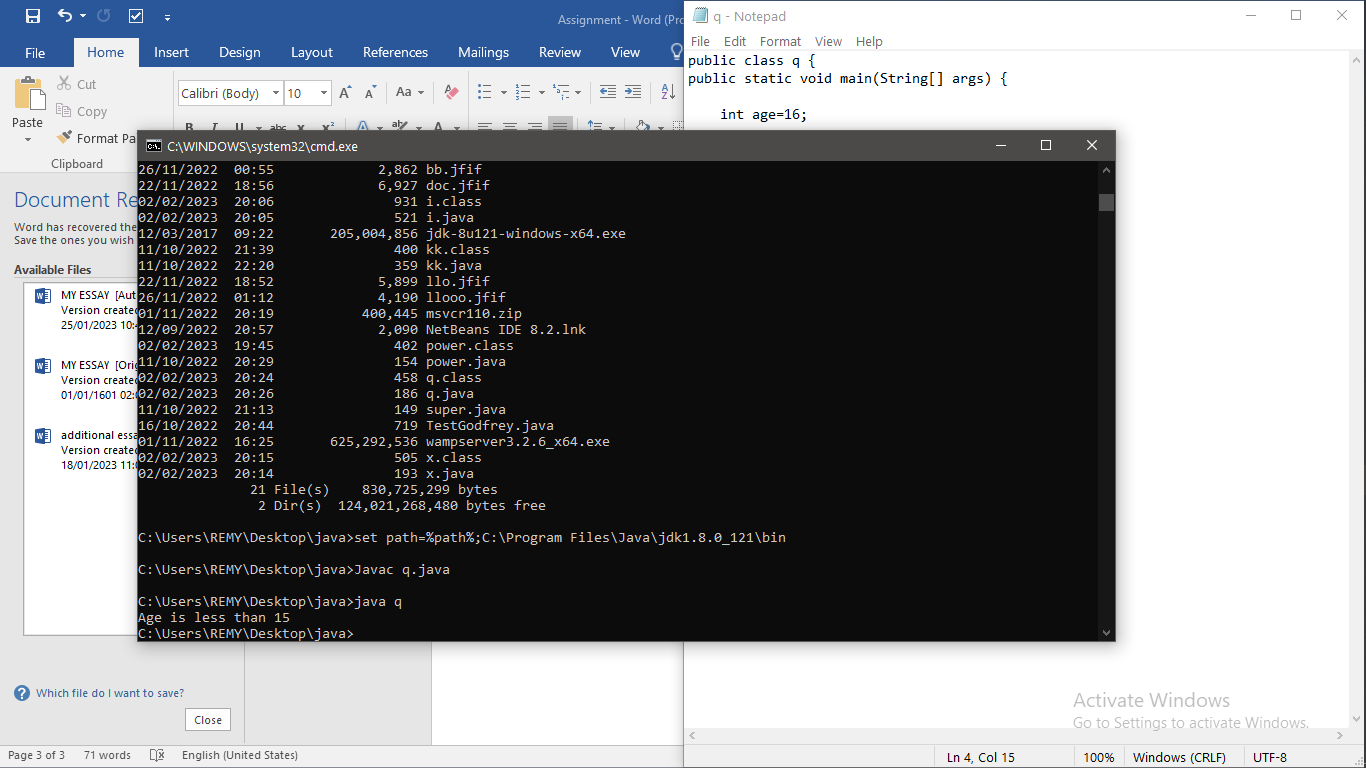
if(age>22){

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

}

}

}



4. class Employee{

float salary=40000;

}

class Programmer extends Employee{

int bonus=10000;

public static void main(String args[]){

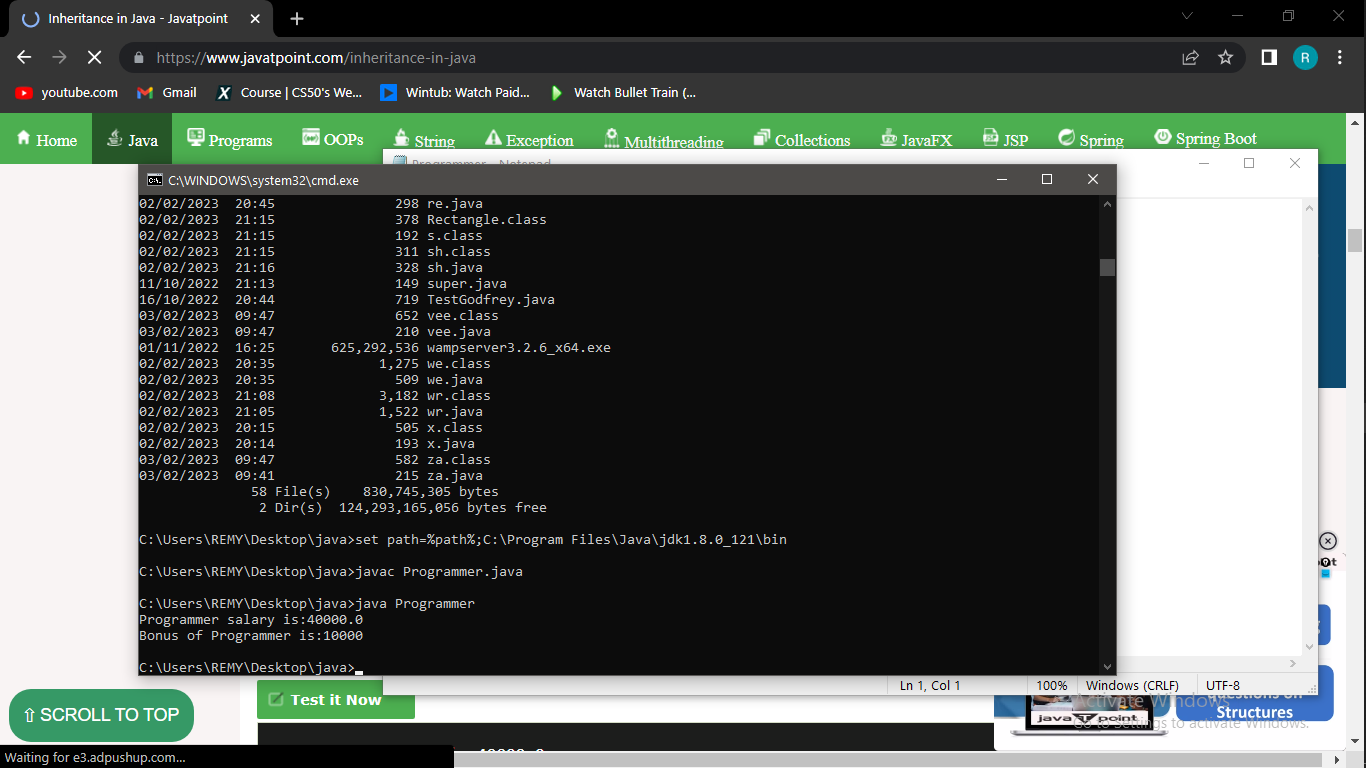
Programmer p=new Programmer();

System.out.println("Programmer salary is:"+p.salary);

System.out.println("Bonus of Programmer is:"+p.bonus);

}

}



5.

class Animal{

void eat(){System.out.println("eating...");}

}

class Dog extends Animal{

void bark(){System.out.println("barking...");}

}

class TestInheritance{

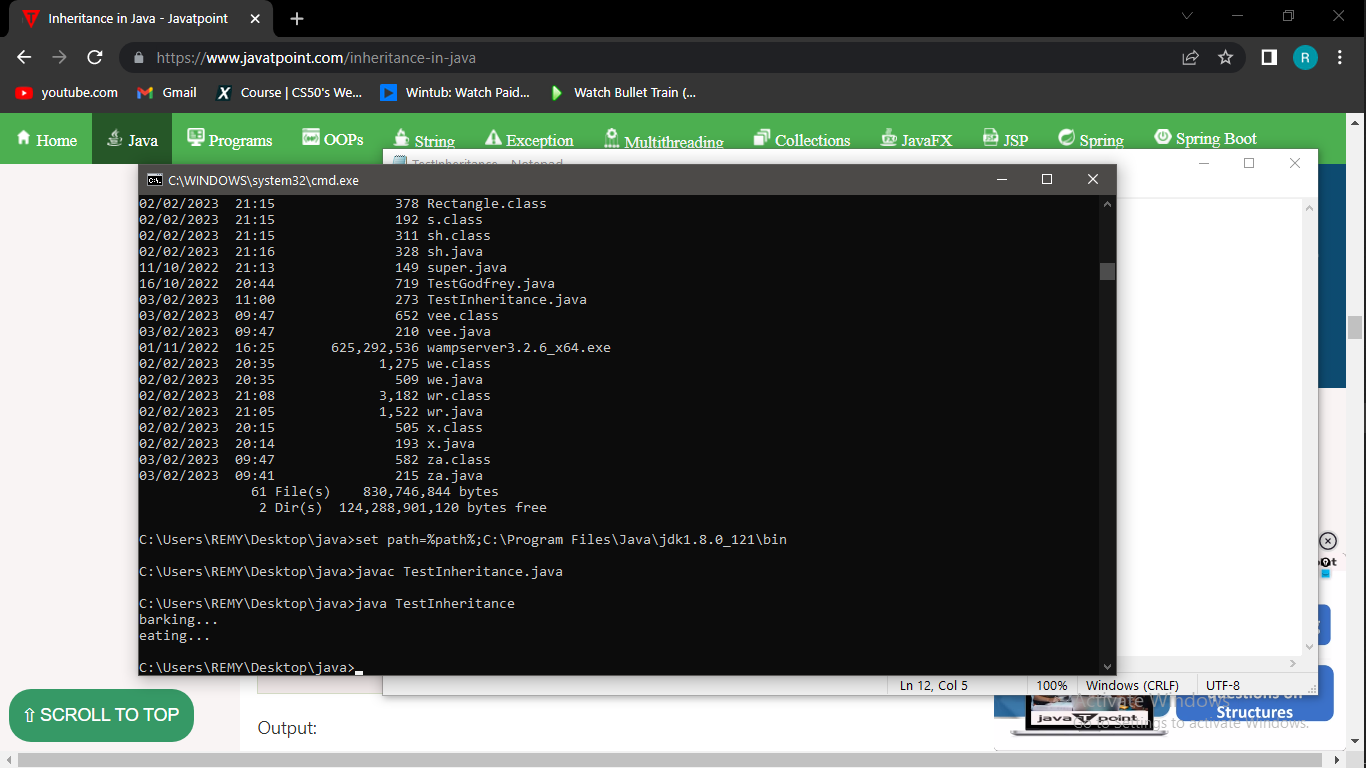
public static void main(String args[]){

Dog d=new Dog();

d.bark();

d.eat();

}}



6.

class Animal{

void eat(){System.out.println("eating...");}

}

class Dog extends Animal{

void bark(){System.out.println("barking...");}

}

class BabyDog extends Dog{

void weep(){System.out.println("weeping...");}

}

class TestInheritance2{

public static void main(String args[]){

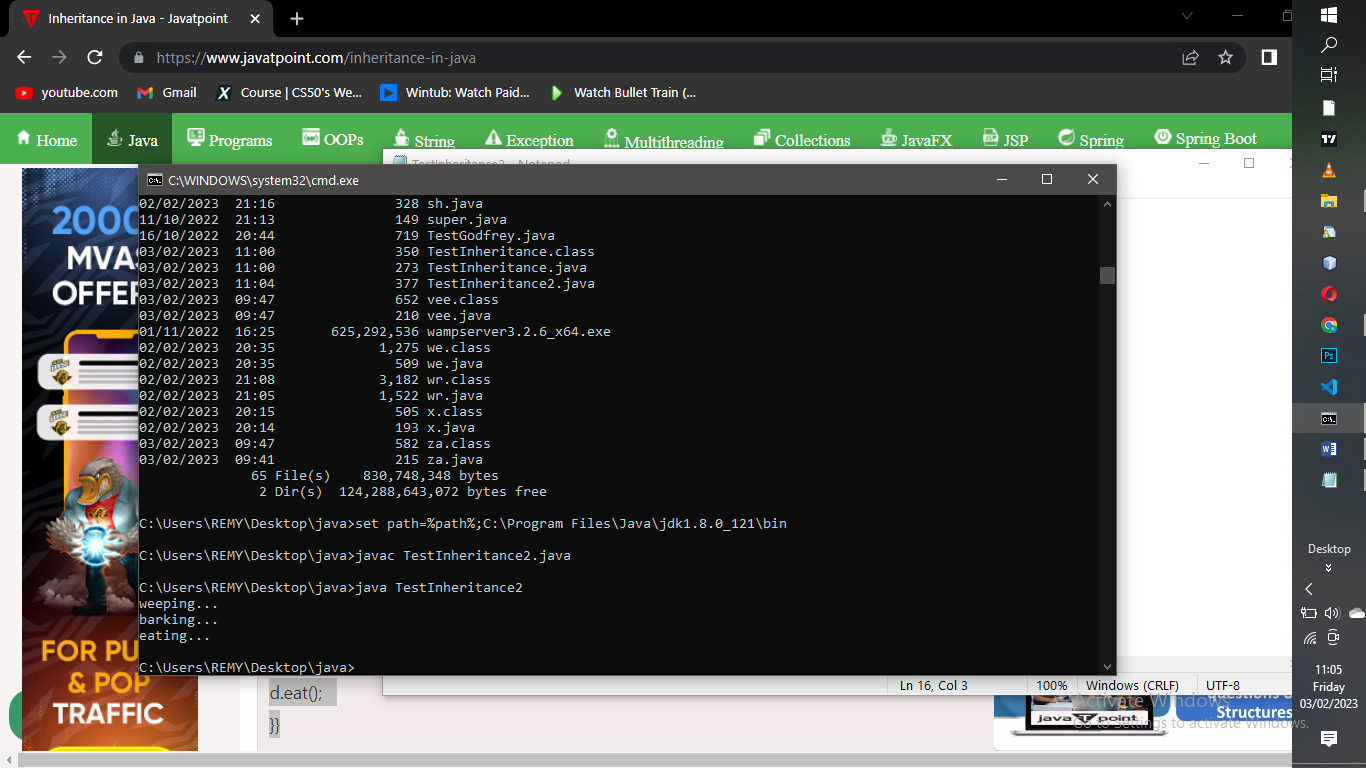
BabyDog d=new BabyDog();

d.weep();

d.bark();

d.eat();

}}



7.

class Adder{

static int add(int a,int b){return a+b;}

static int add(int a,int b,int c){return a+b+c;}

}

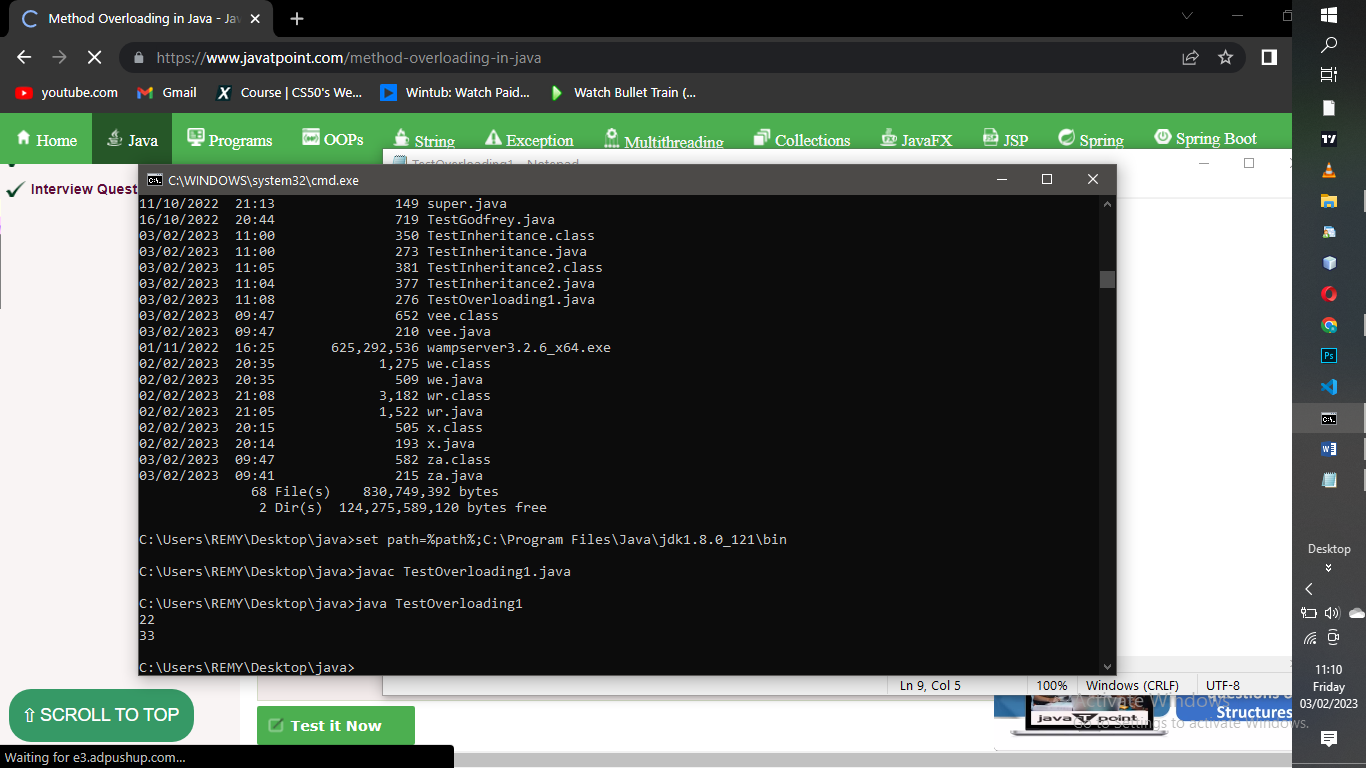
class TestOverloading1{

public static void main(String[] args){

System.out.println(Adder.add(11,11));

System.out.println(Adder.add(11,11,11));

}}



8.

class Adder{

static int add(int a, int b){return a+b;}

static double add(double a, double b){return a+b;}

}

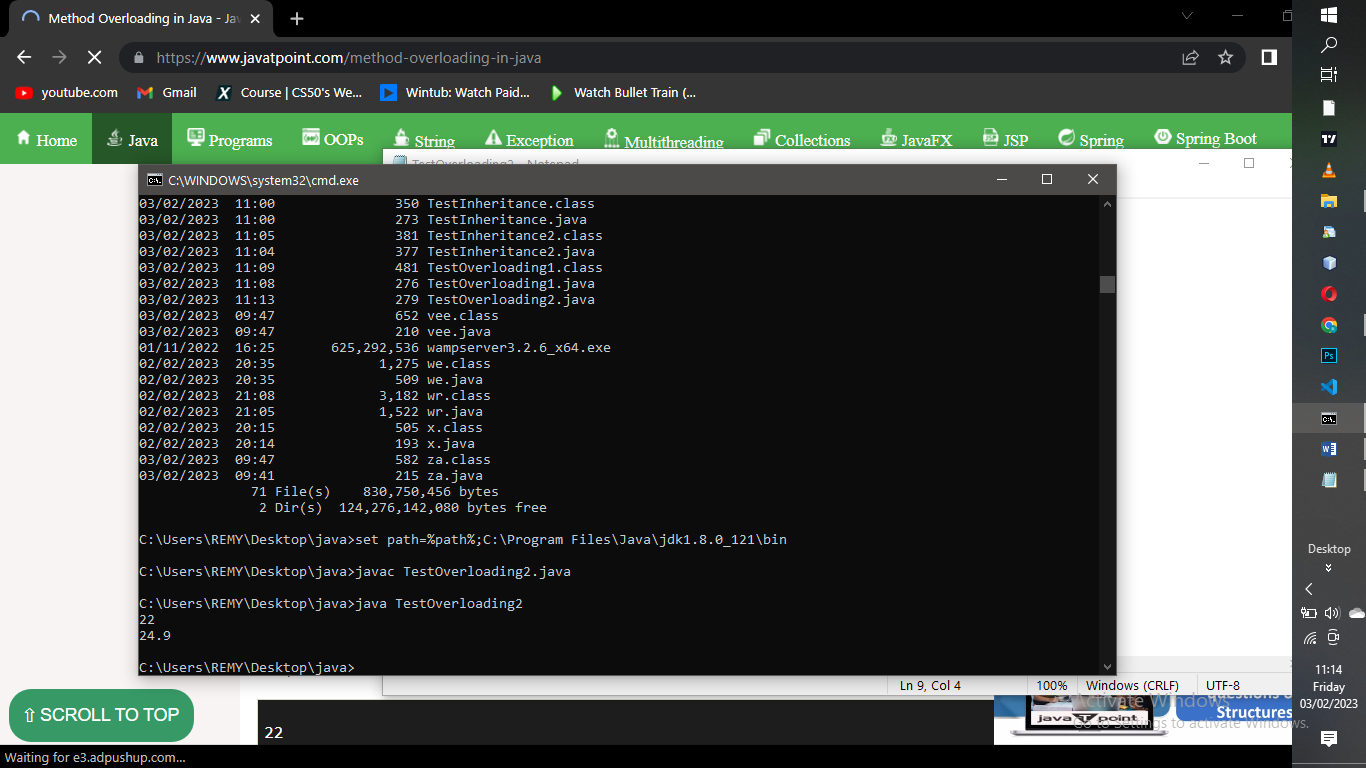
class TestOverloading2{

public static void main(String[] args){

System.out.println(Adder.add(11,11));

System.out.println(Adder.add(12.3,12.6));

}}



9.

class Adder{

static int add(int a,int b){return a+b;}

static double adder(int a,int b){return a+b;}

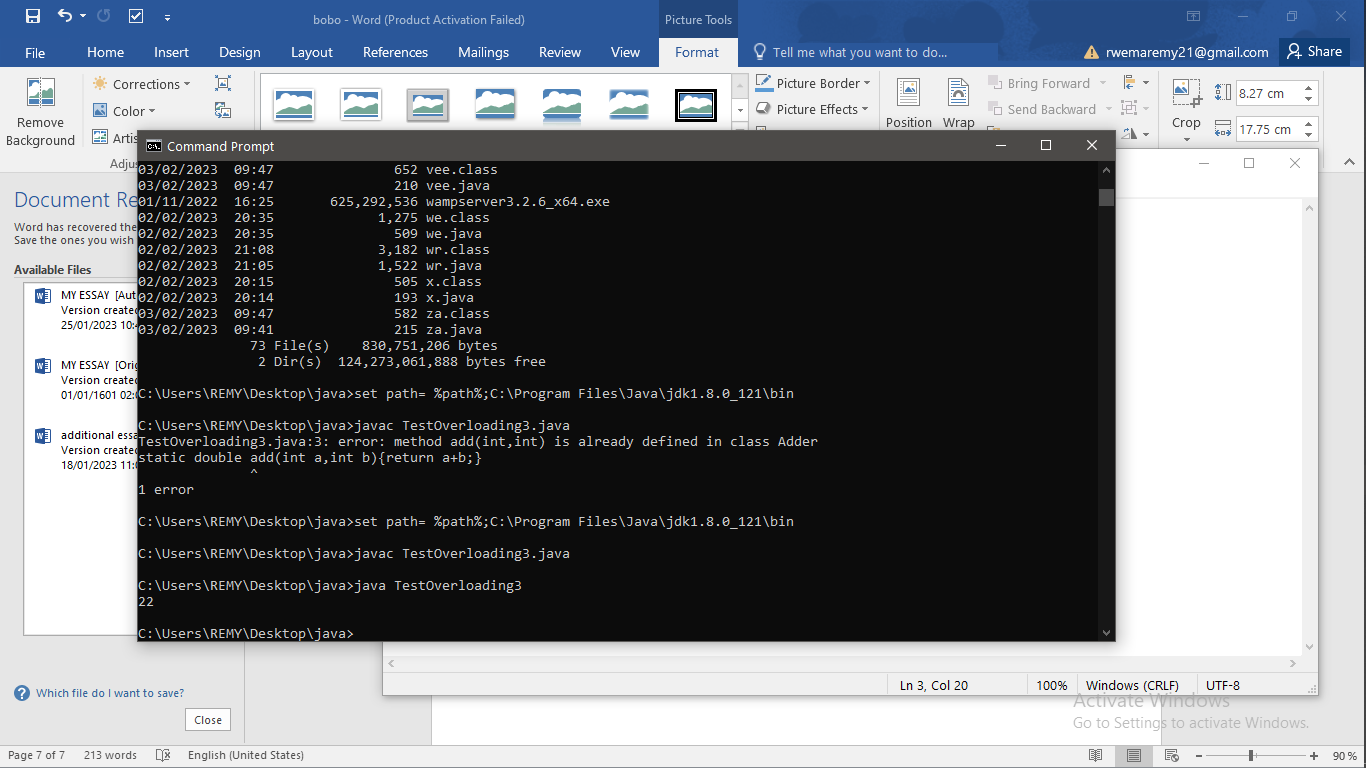
}

class TestOverloading3{

public static void main(String[] args){

System.out.println(Adder.add(11,11));//ambiguity

}}



10.string replace

public class za{

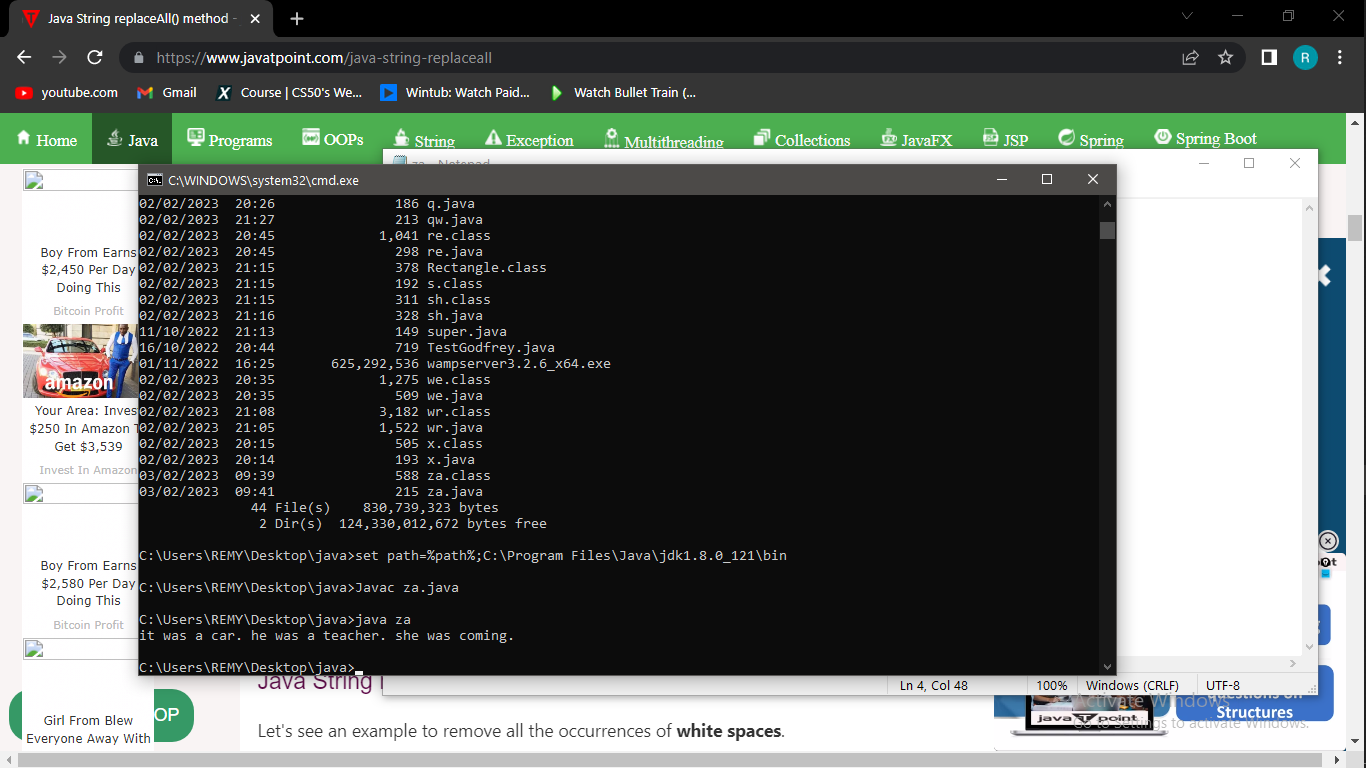
public static void main(String args[]){

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";

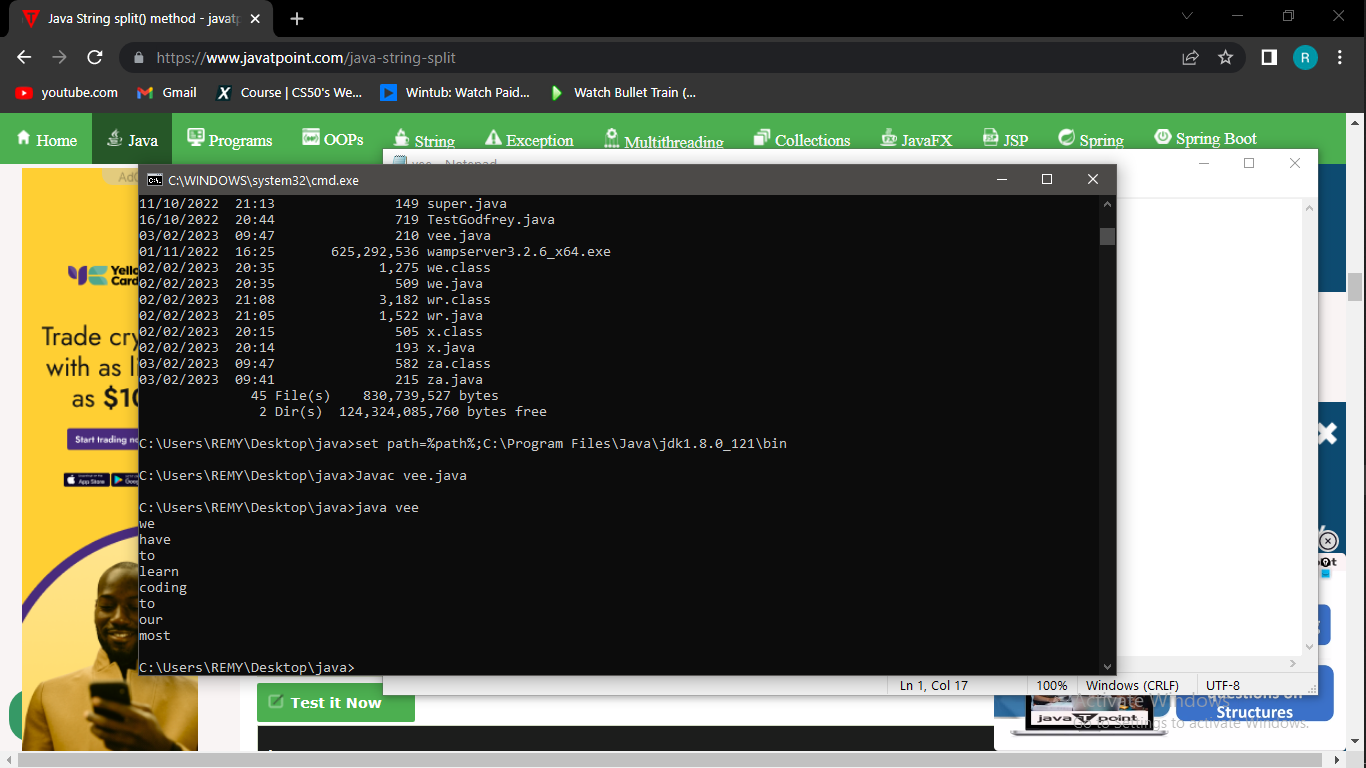
String[] words=s1.split("\\s");

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();

}

}

