Assignment

1.

class Student{

int id;

String name;

}

class TestStudent2{

public static void main(String args[]){

Student s1=new Student();

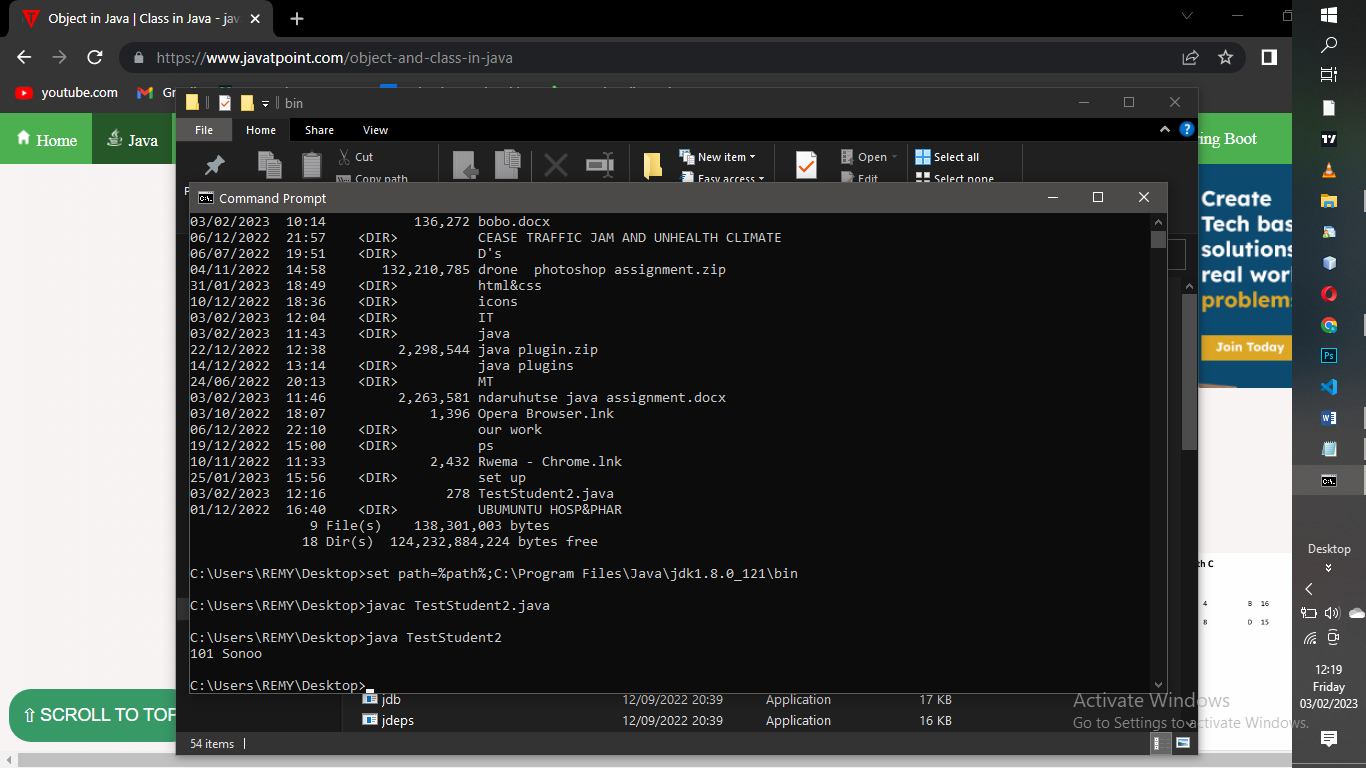
s1.id=101;

s1.name="Sonoo";

System.out.println(s1.id+" "+s1.name);//printing members with a white space

}

}



2.

public class Addition

{

public static void main(String[] args)

{

int a = 19;

int b = 5;

int c = add(a, b);

System.out.println("The sum of a and b is= " + c);

}

public static int add(int n1, int n2)

{

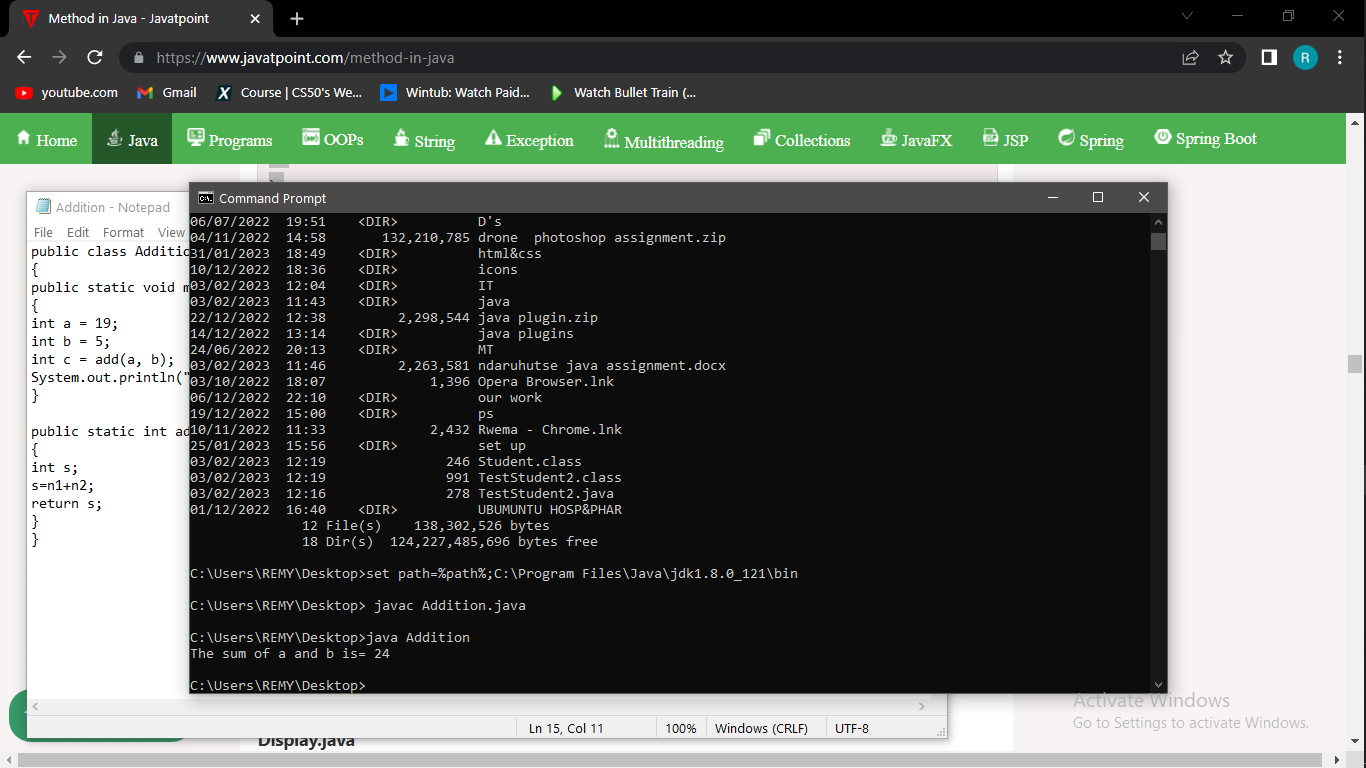
int s;

s=n1+n2;

return s;

}

}



3. class Bike1{

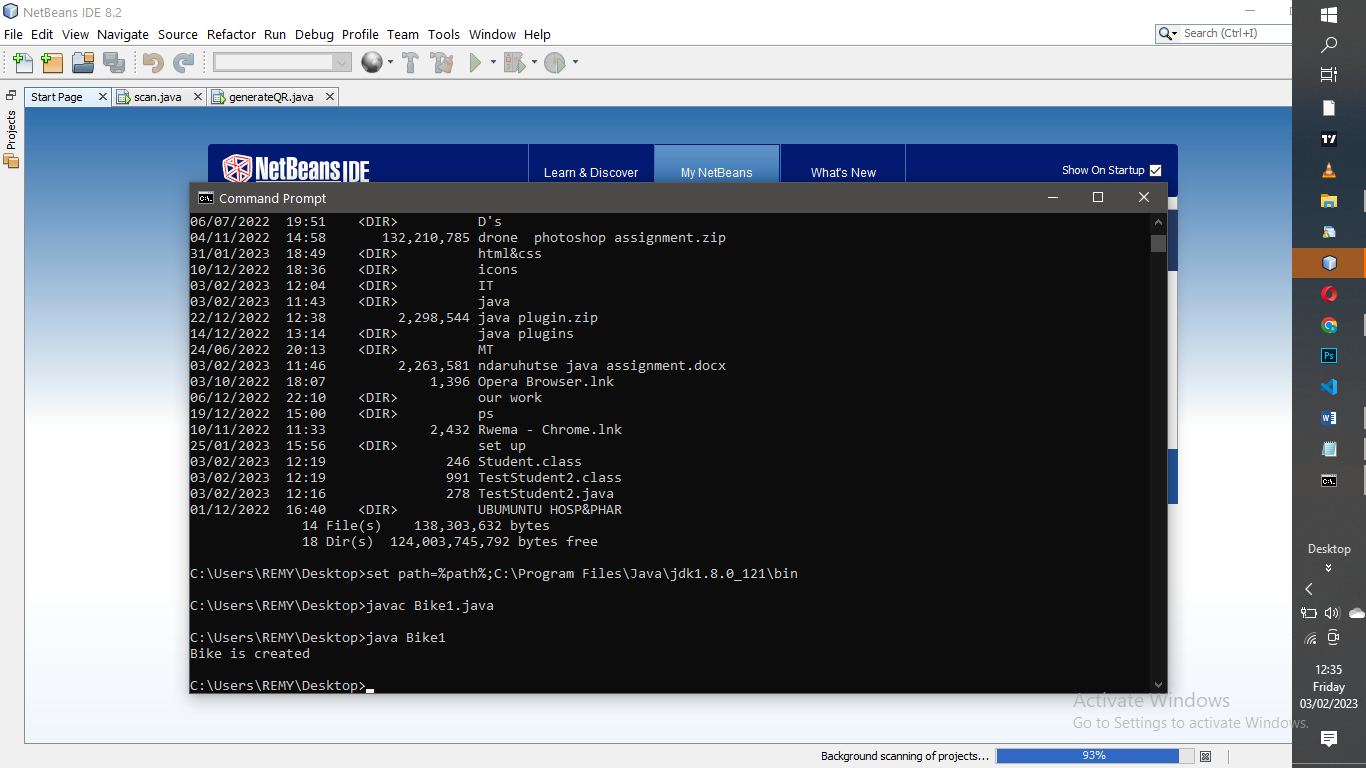
Bike1(){System.out.println("Bike is created");}

public static void main(String args[]){

Bike1 b=new Bike1();

}

}



4.

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[]){

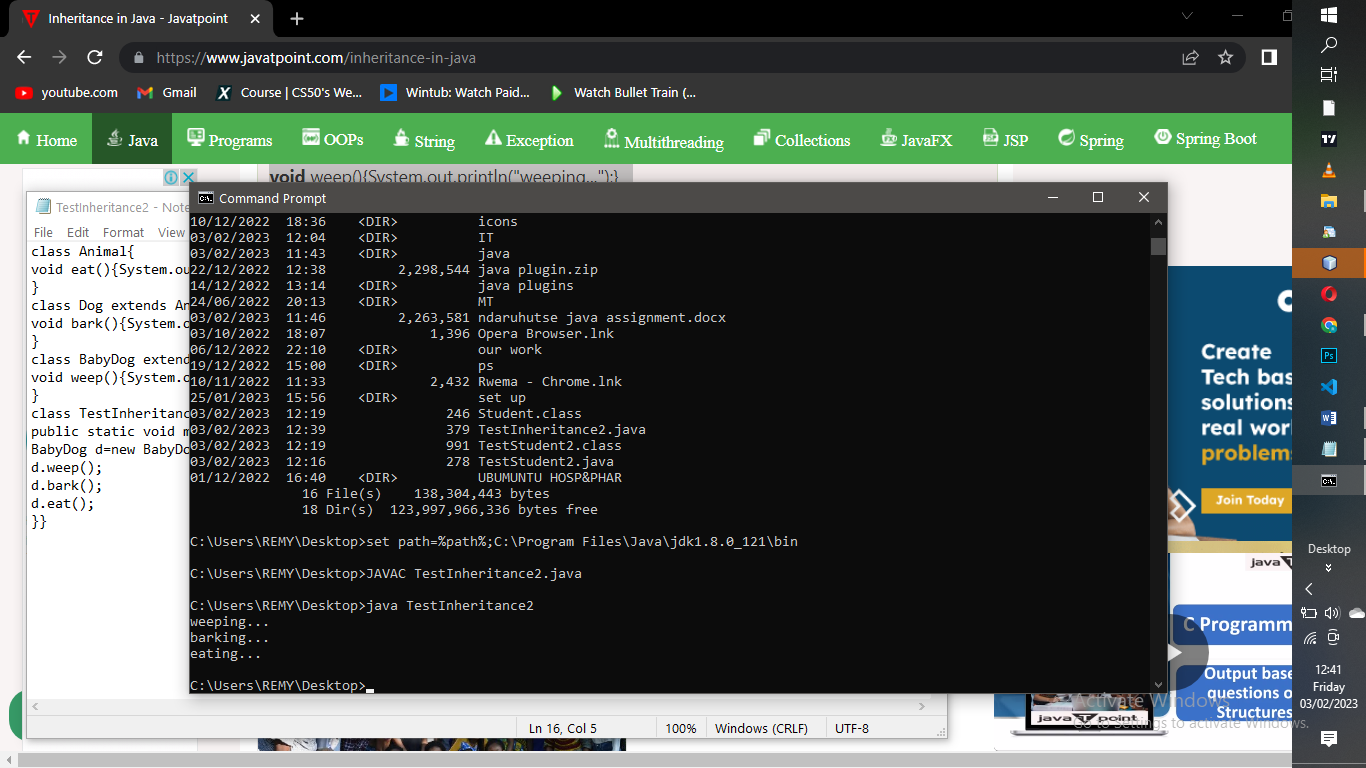
BabyDog d=new BabyDog();

d.weep();

d.bark();

d.eat();

}}



5.

class Animal{

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

}

class Dog extends Animal{

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

}

class Cat extends Animal{

void meow(){System.out.println("meowing...");}

}

class TestInheritance3{

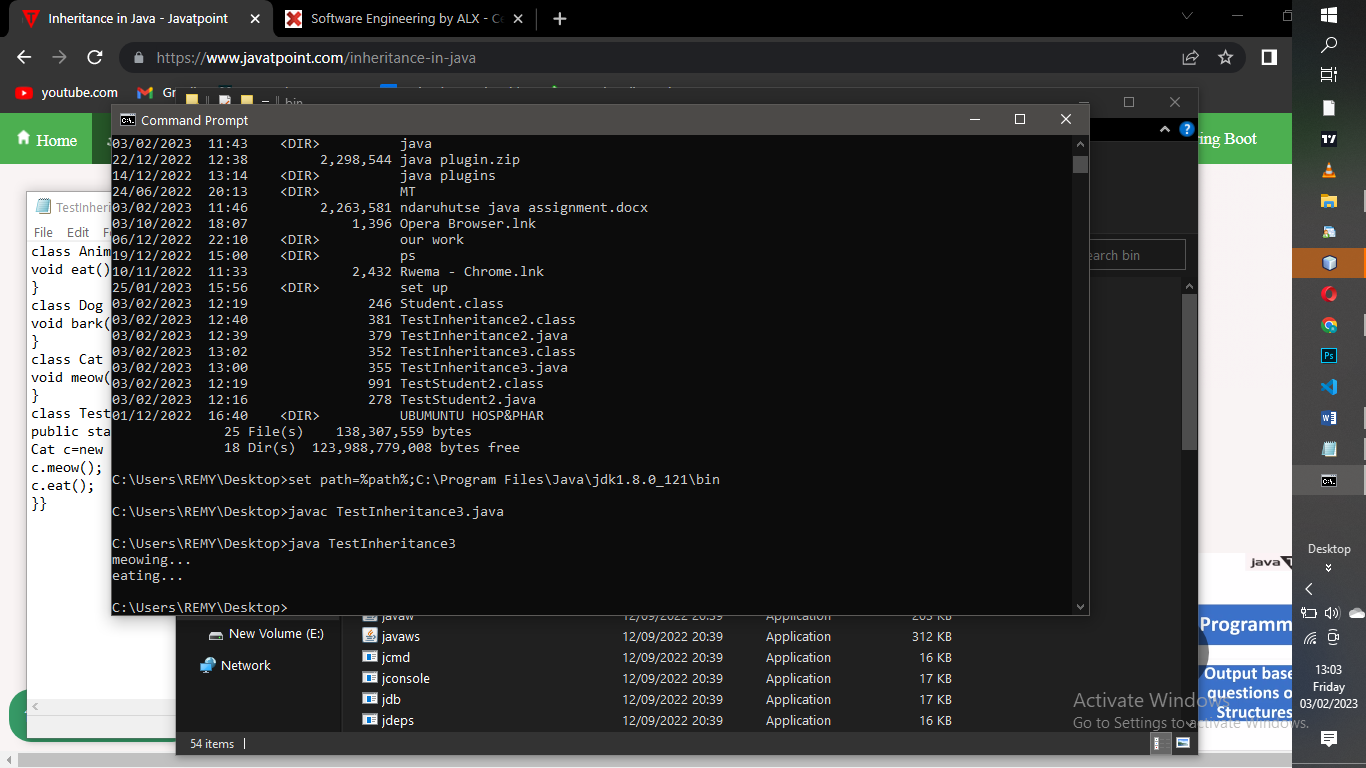
public static void main(String args[]){

Cat c=new Cat();

c.meow();

c.eat();

}}



6.

class Employee{

float salary=40000;

}

class Programmer extends Employee{

int bonus=10000;

public static void main(String args[]){

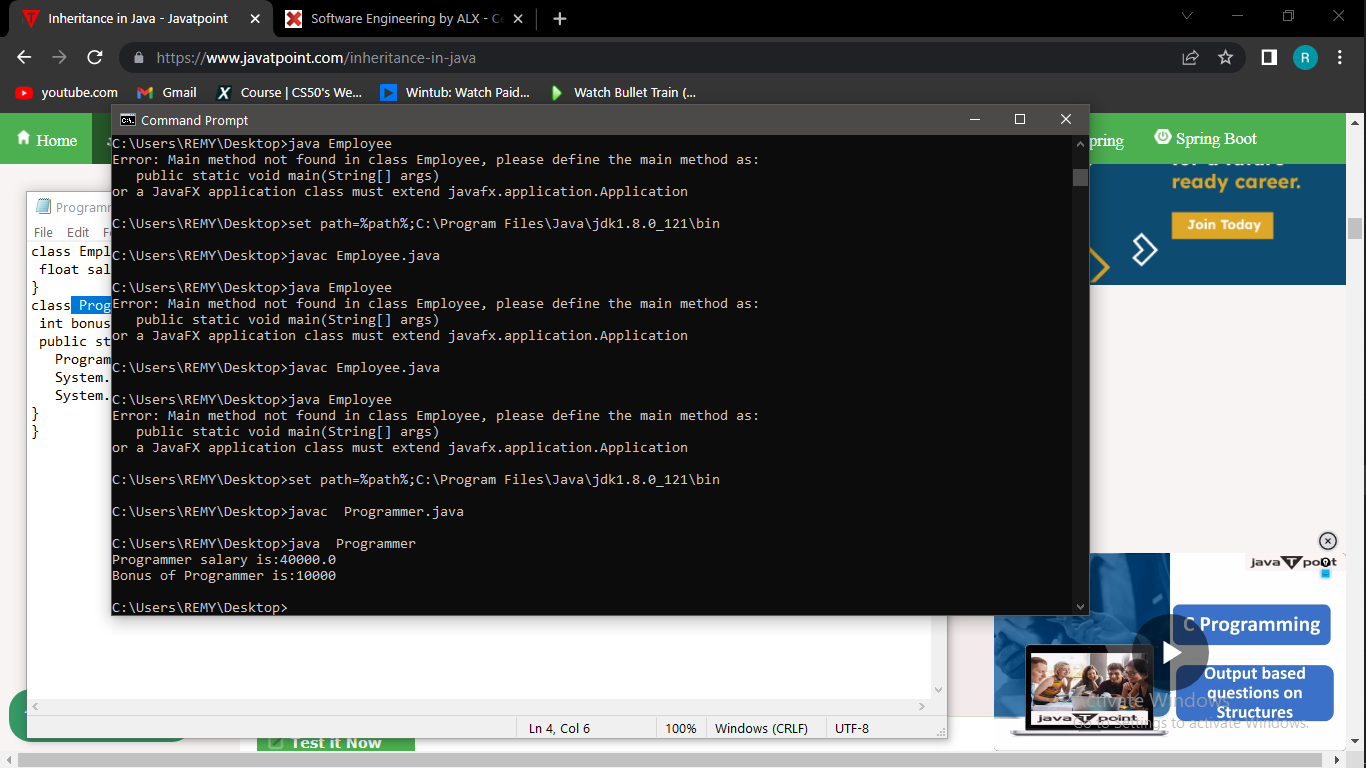
Programmer p=new Programmer();

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

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

}

}



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;}

}

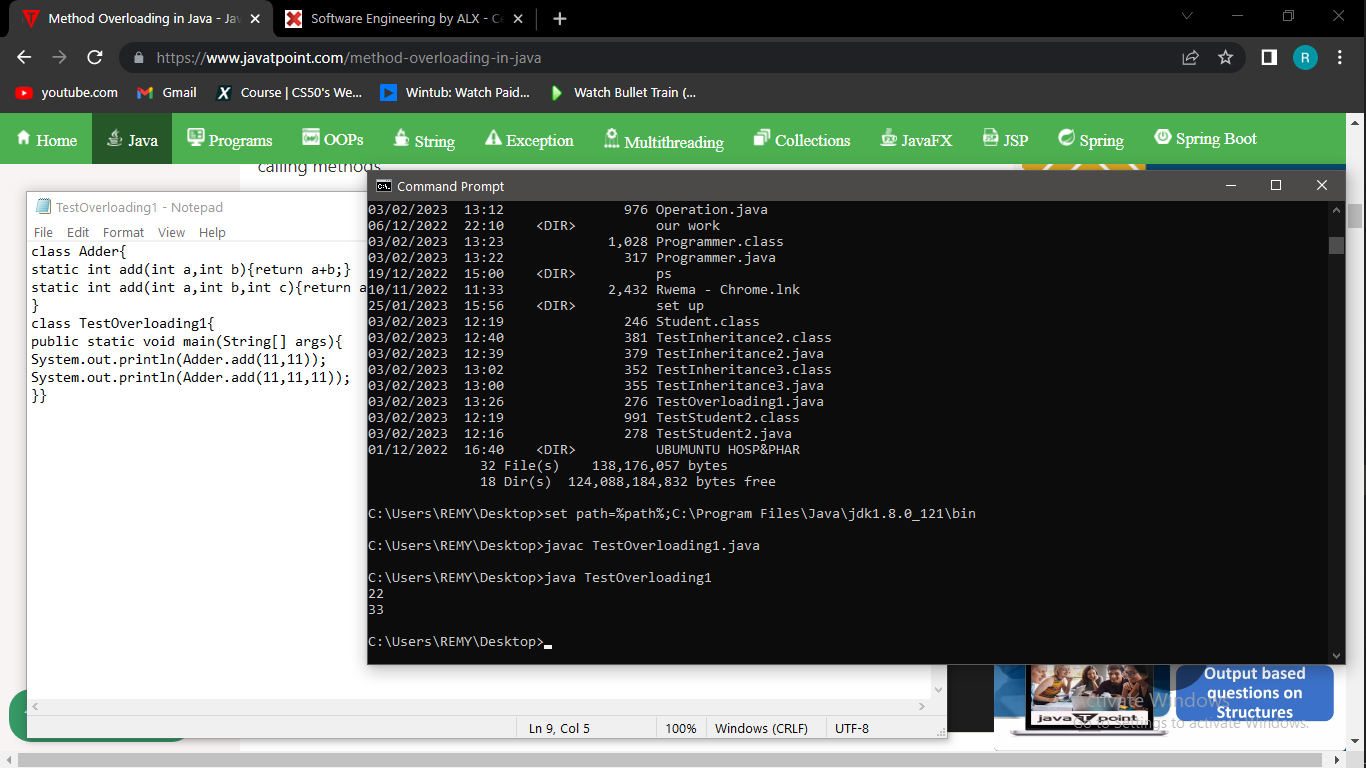
class TestOverloading1{

public static void main(String[] args){

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

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

}}



8.

class Vehicle{

//defining a method

void run(){System.out.println("Vehicle is running");}

}

//Creating a child class

class Bike2 extends Vehicle{

//defining the same method as in the parent class

void run(){System.out.println("Bike is running safely");}

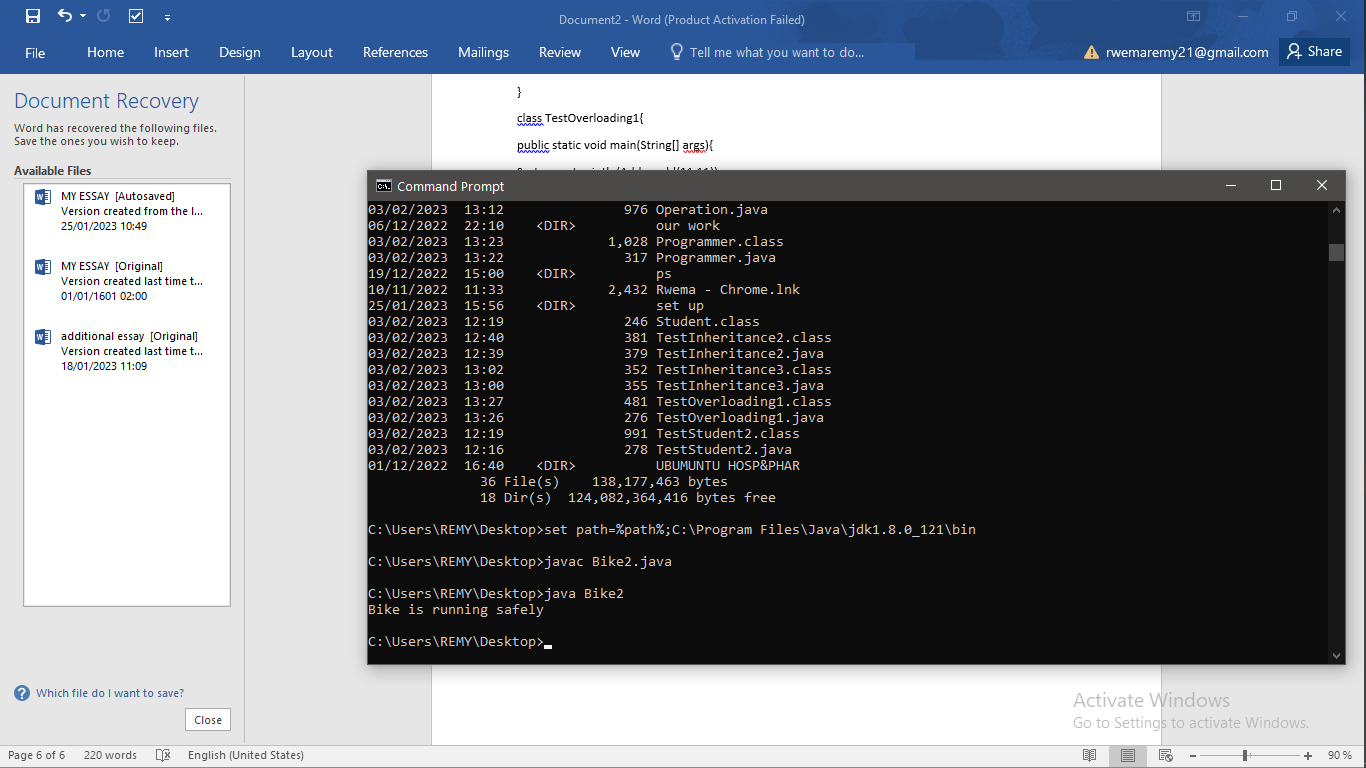
public static void main(String args[]){

Bike2 obj = new Bike2();//creating object

obj.run();//calling method

}

}



9.

class Bike{

final void run(){System.out.println("running...");}

}

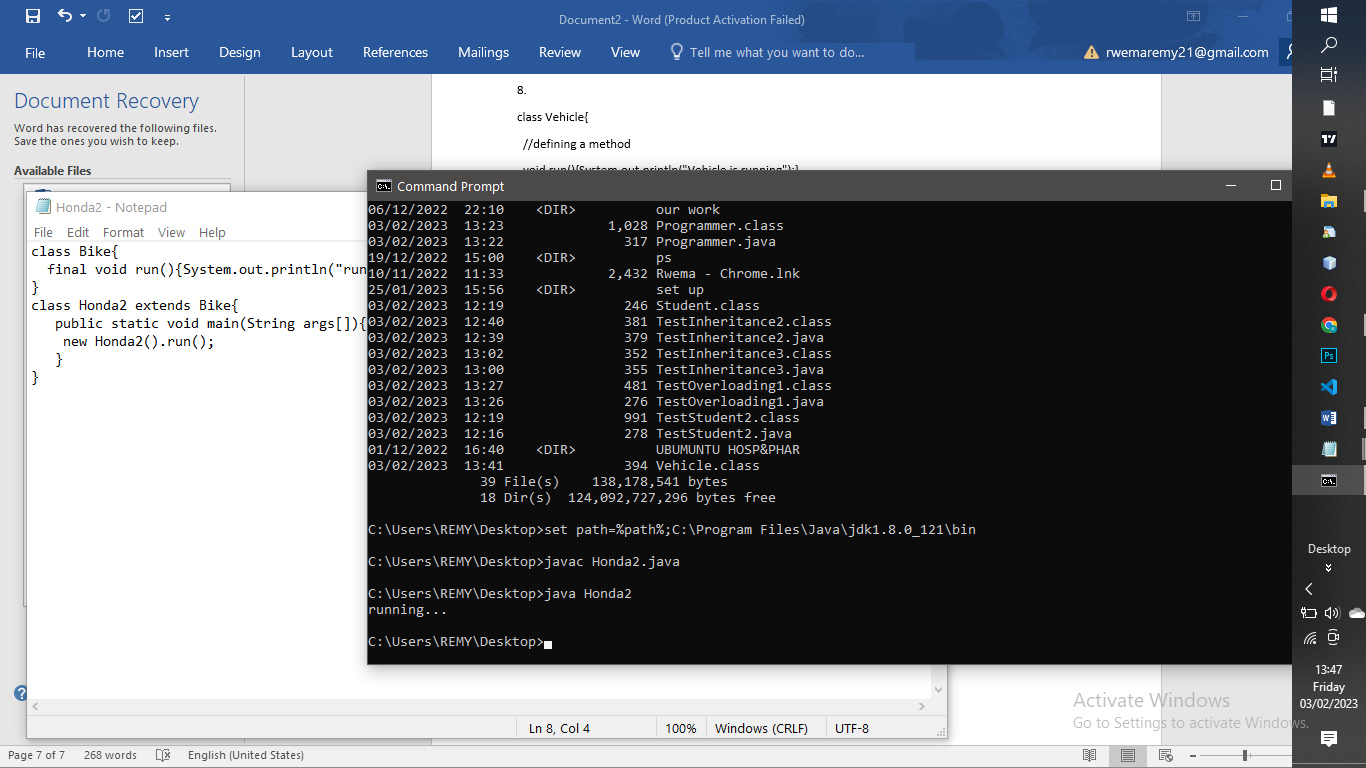
class Honda2 extends Bike{

public static void main(String args[]){

new Honda2().run();

}

}



10. abstract class Bike{

abstract void run();

}

class Honda4 extends Bike{

void run(){System.out.println("running safely");}

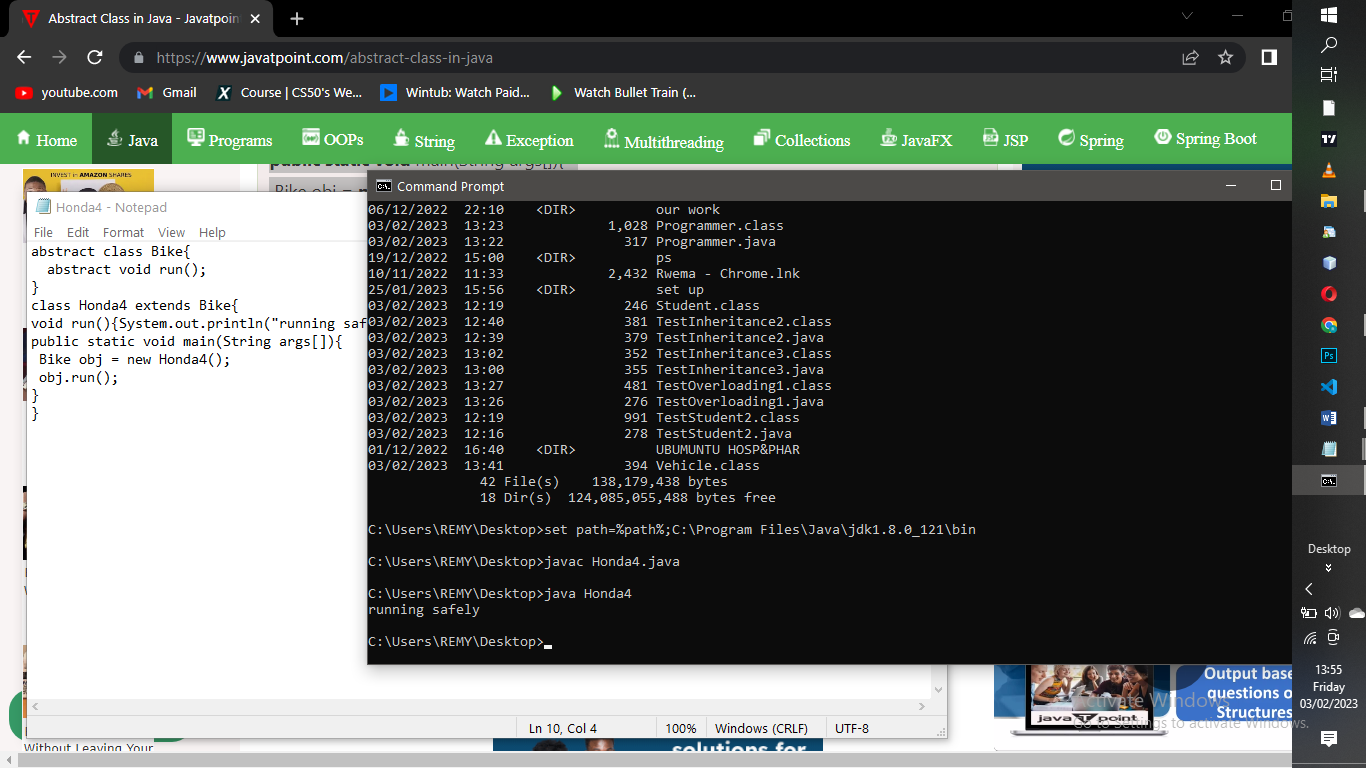
public static void main(String args[]){

Bike obj = new Honda4();

obj.run();

}

}



11.

interface Bank{

float rateOfInterest();

}

class SBI implements Bank{

public float rateOfInterest(){return 9.15f;}

}

class PNB implements Bank{

public float rateOfInterest(){return 9.7f;}

}

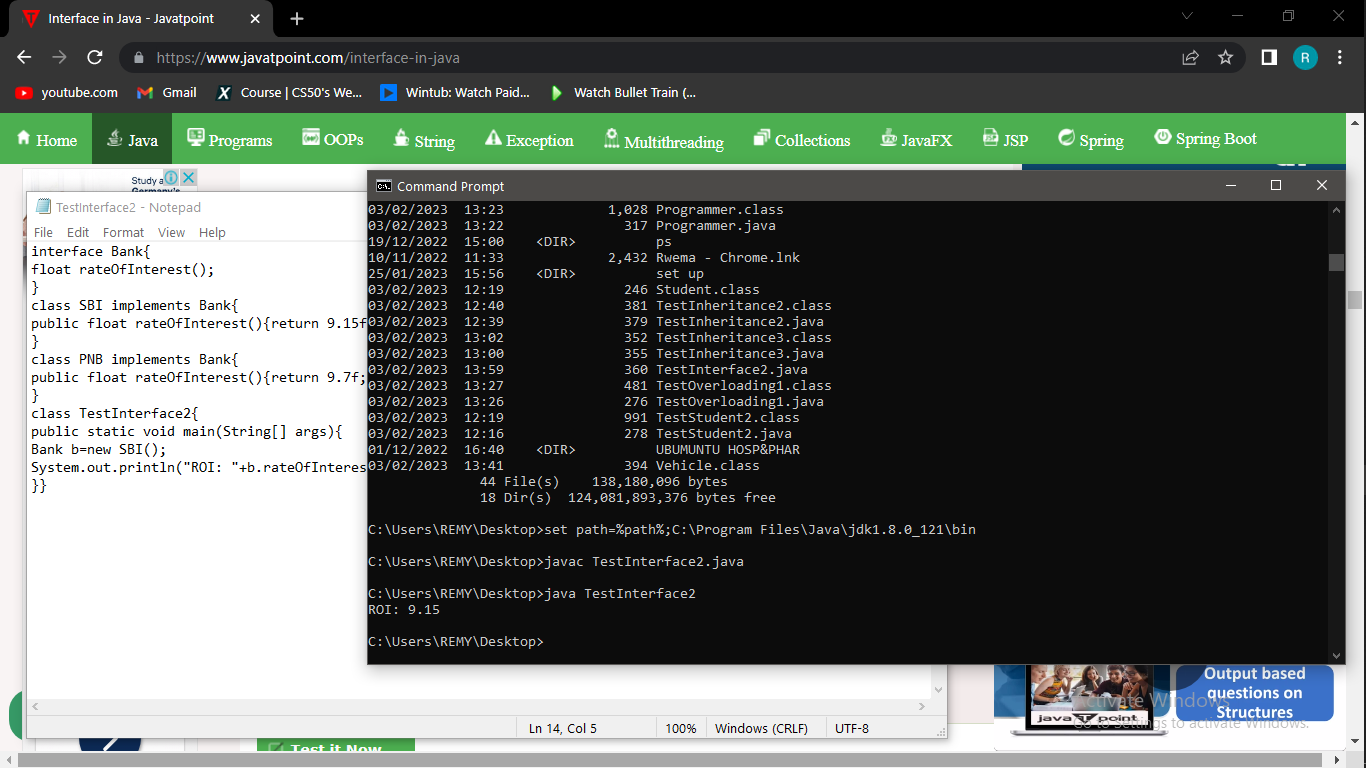
class TestInterface2{

public static void main(String[] args){

Bank b=new SBI();

System.out.println("ROI: "+b.rateOfInterest());

}}



12.

interface printable{

void print();

}

class A6 implements printable{

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

public static void main(String args[]){

A6 obj = new A6();

obj.print();

}

}

