MUREKATETE BERTHE

JAVA ASSGNMENT

Control statement experiment 1.if condition

public class berthe {

public static void main(String[] args) {

//defining an 'age' variable

int age=30;

//checking the age

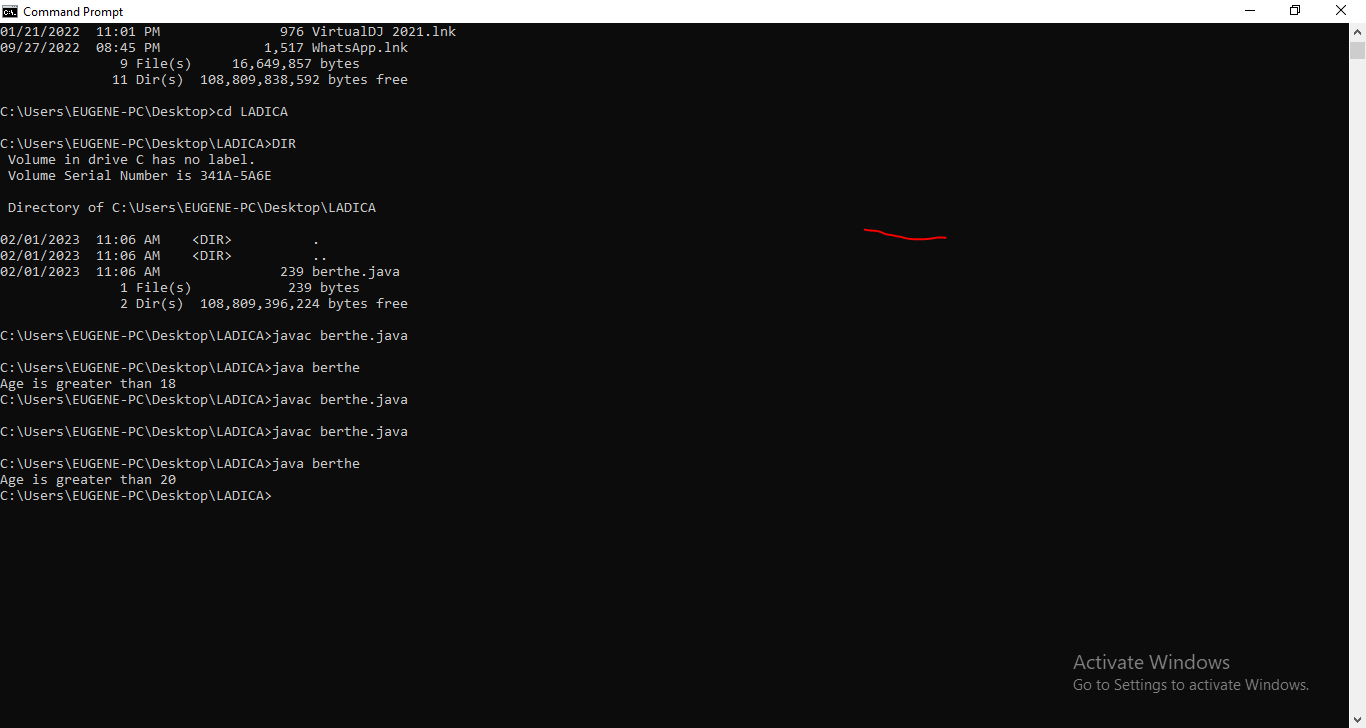
if(age>20){

System.out.print("Age is greater than 20");

}

}

}



Java if-else Statement

Example

public class berthe{

public static void main(String[] args) {

int number=13;

if(number%2==0){

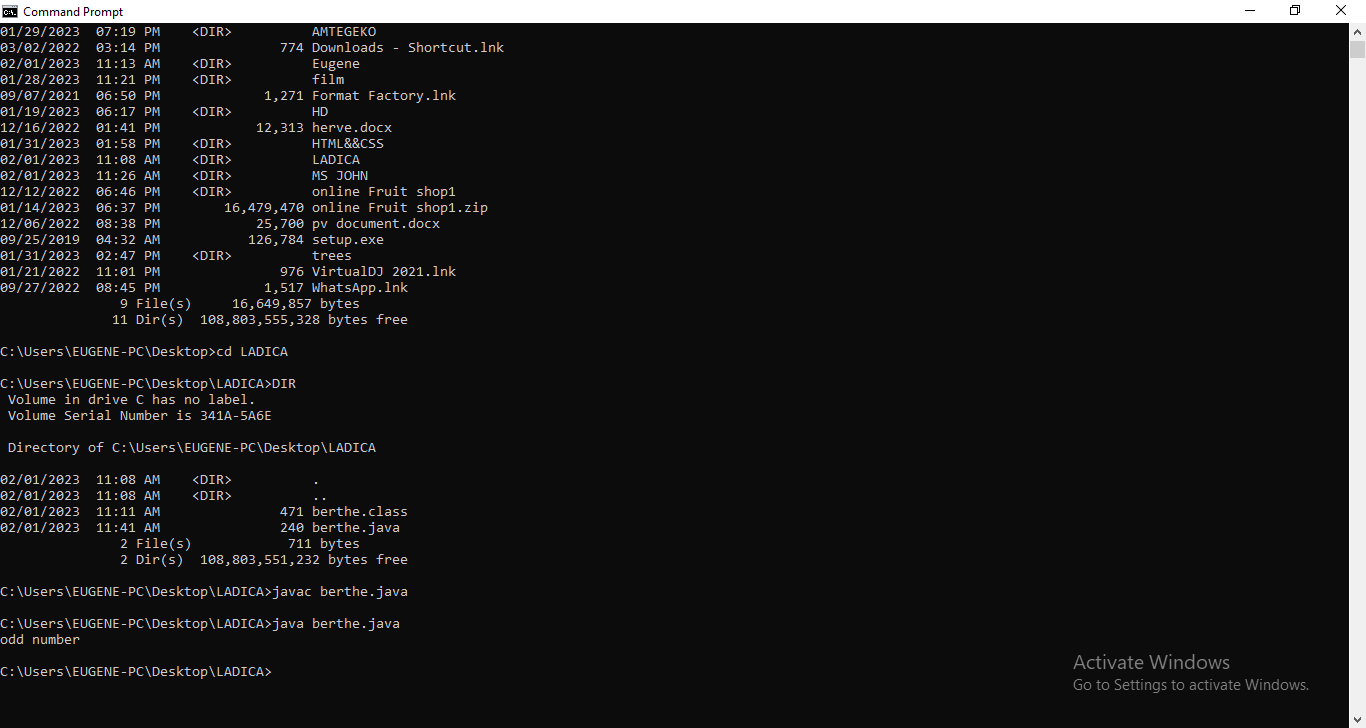
System.out.println("even number");

}else{

System.out.println("odd number");

}

}



EXPERIMENT 3

public class PositiveNegative {

public static void main(String[] args) {

int number=-15;

if(number>0){

System.out.println("POSITIVE");

}else if(number<0){

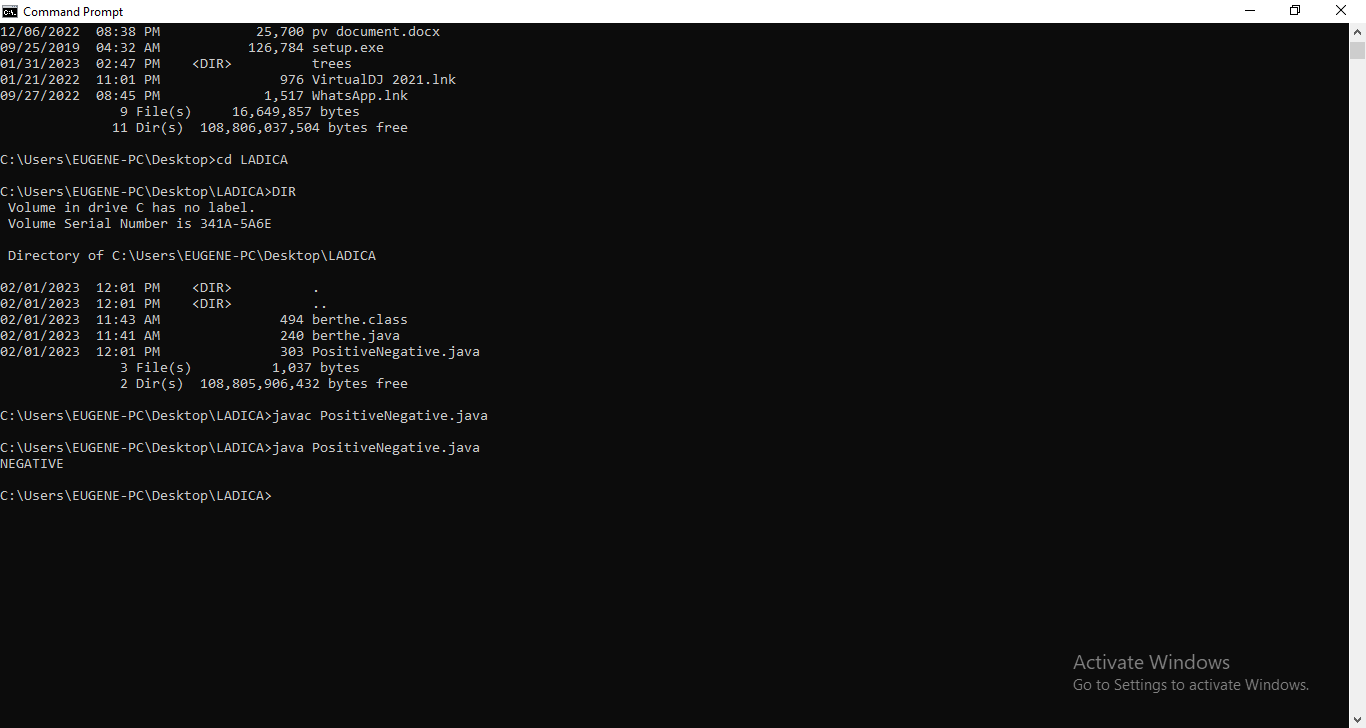
System.out.println("NEGATIVE");

}else{

System.out.println("ZERO");

}

}



OBJECT AND CLASS IN JAVA

EXPERIMENT 1

class Student{

int id;

String name;

public static void main(String args[]){

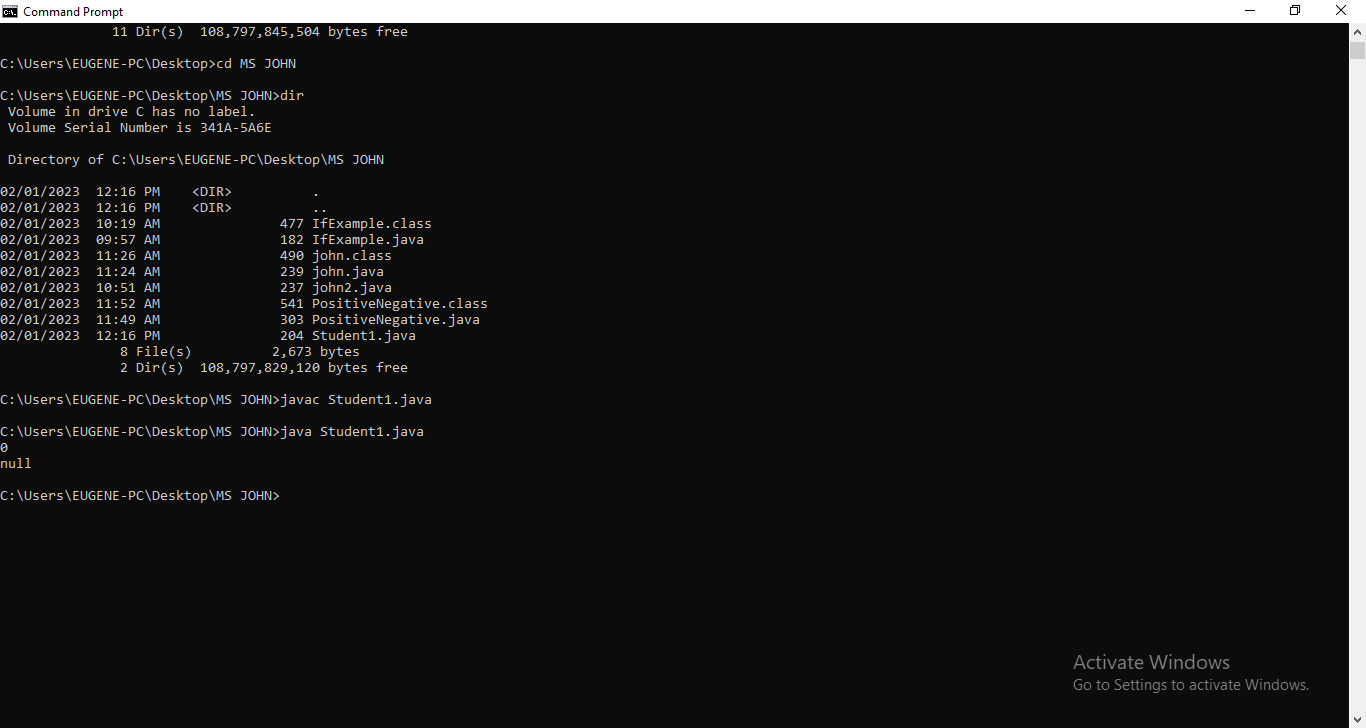
Student s1=new Student();

System.out.println(s1.id);

System.out.println(s1.name);

}

}



EXPERIMENT 2

class Student{

int id;

String name;

}

class TestStudent3{

public static void main(String args[]){

Student s1=new Student();

Student s2=new Student();

s1.id=101;

s1.name="Sonoo";

s2.id=102;

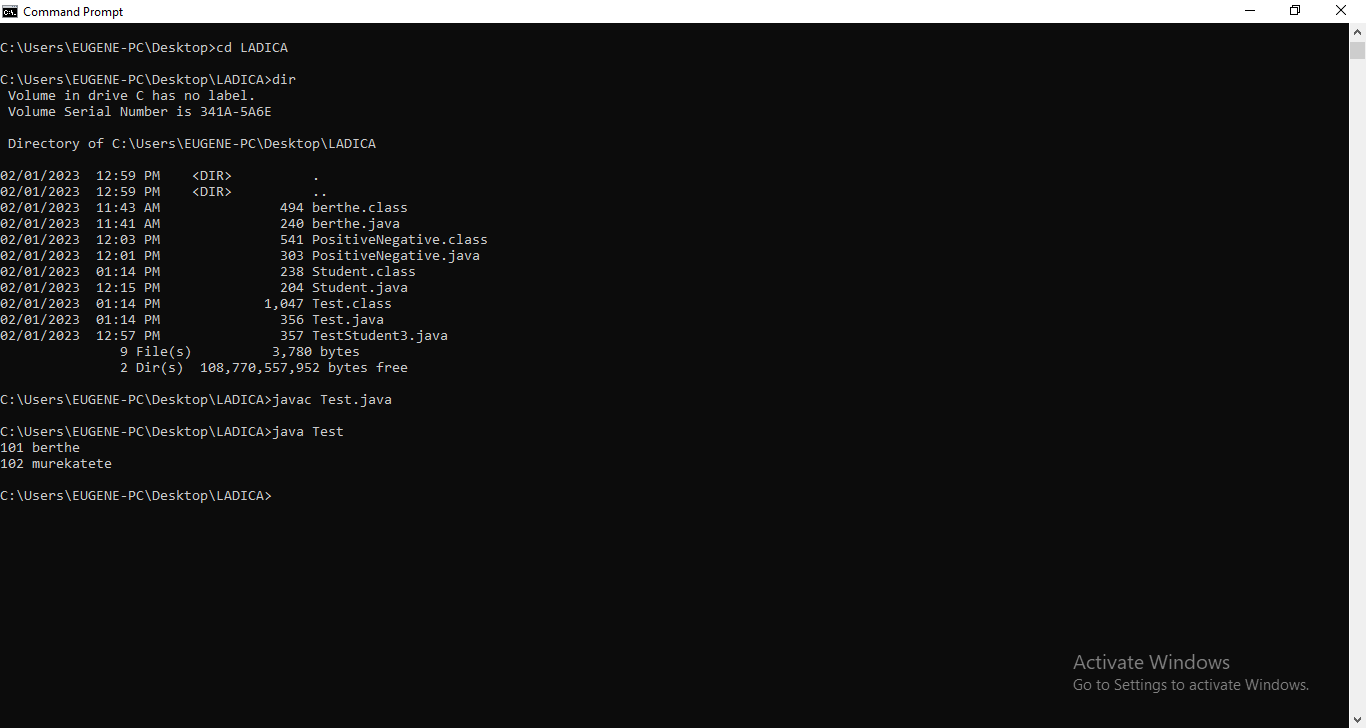
s2.name="Amit";

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

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

}

}



EXPERIMENT 3

class Rectangle{

int length;

int width;

void insert(int l, int w){

length=l;

width=w;

}

void calculateArea(){System.out.println(length\*width);}

}

class Rectangle1{

public static void main(String args[]){

Rectangle r1=new Rectangle();

Rectangle r2=new Rectangle();

r1.insert(10,4);

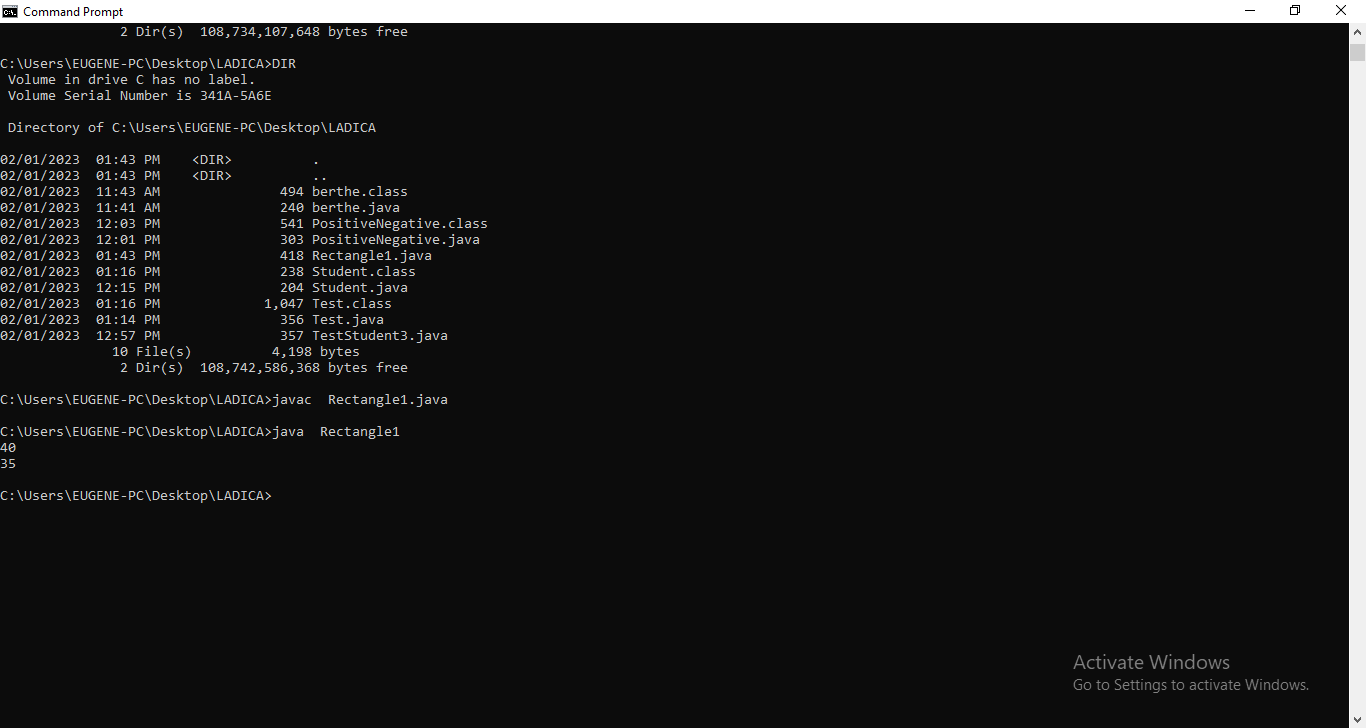
r2.insert(7,5);

r1.calculateArea();

r2.calculateArea();

}

}



# **Inheritance in Java**

EXPERIMENT 1

class Customer{

float salary=80000;

}

class Programmer extends Customer{

int bonus=5000;

public static void main(String args[]){

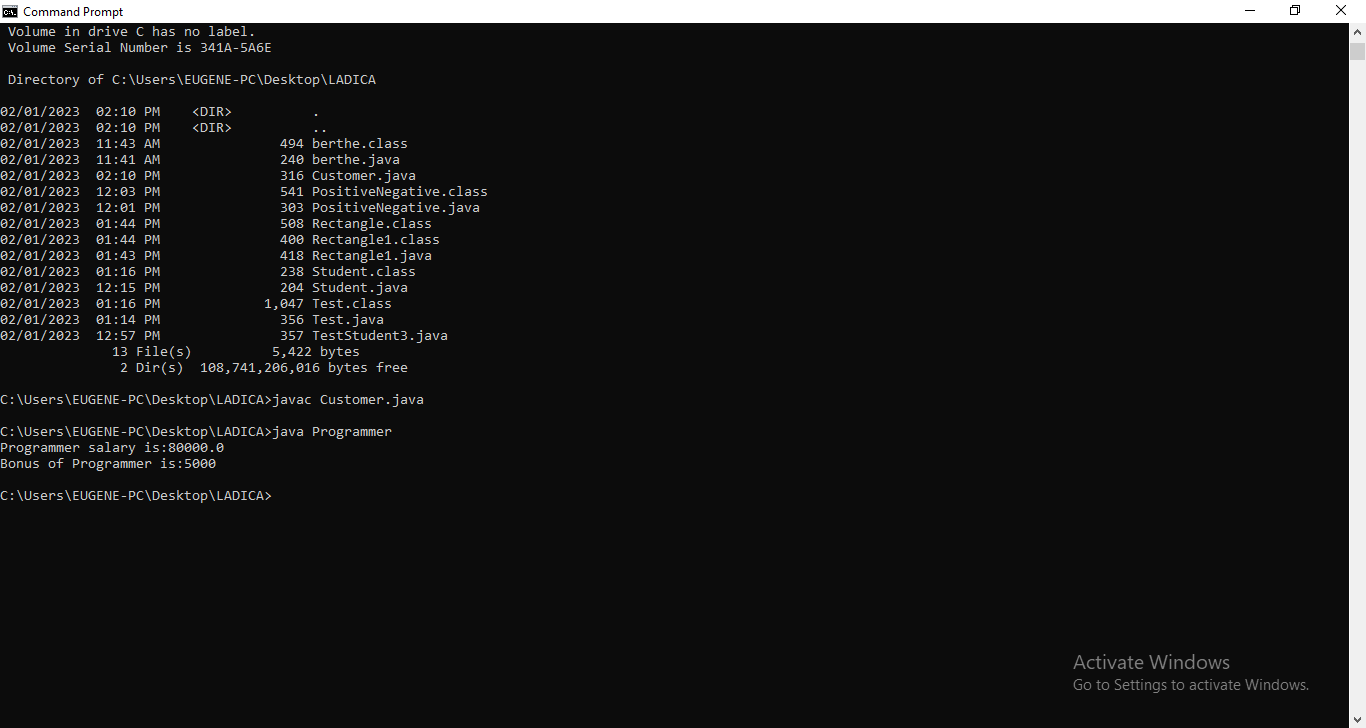
Programmer p=new Programmer();

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

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

}

}



EXPERIMENT 2

class Animal{

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

}

class Dog extends Animal{

void bark()

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

}

class TestInheritance{

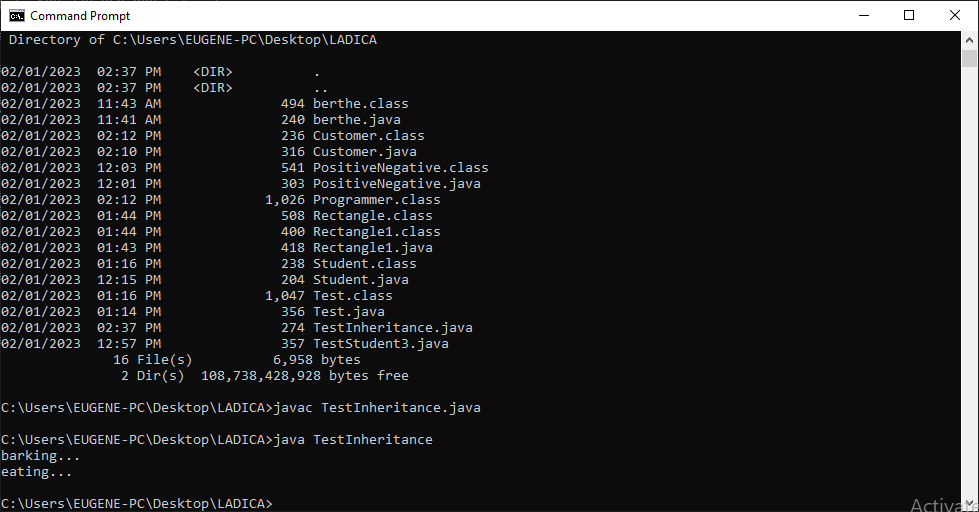
public static void main(String args[]){

Dog d=new Dog();

d.bark();

d.eat();

}}



EXPERIMENT 3

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 Test2{

public static void main(String args[]){

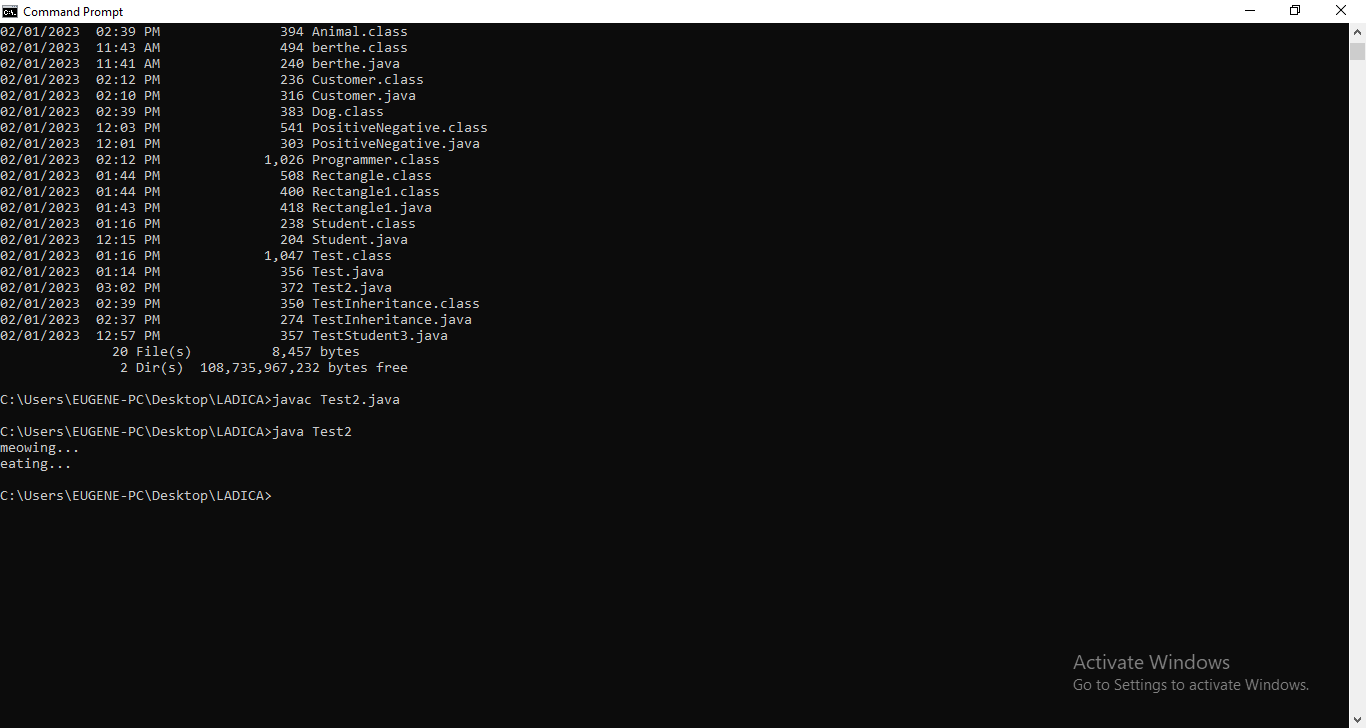
Cat c=new Cat();

c.meow();

c.eat();

//c.bark();//C.T.Error

}}



## Java Polymorphism

Experiment 1

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;}

}

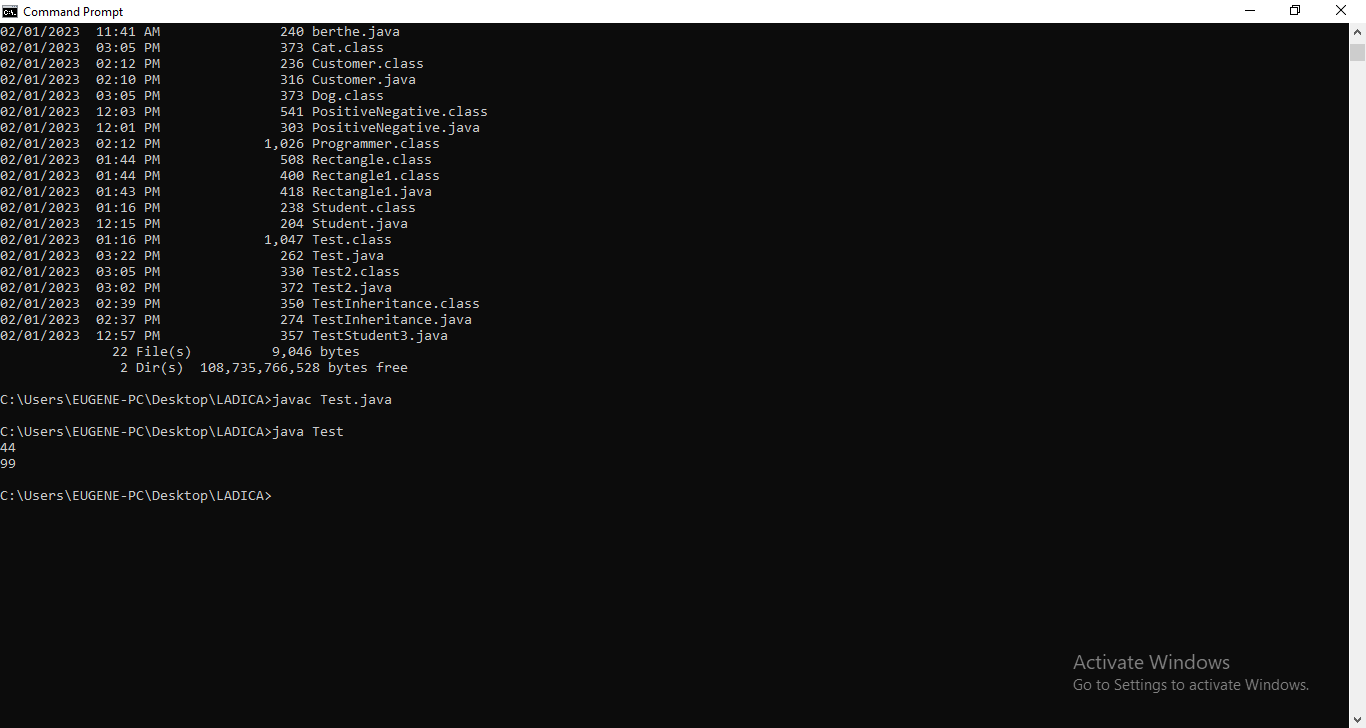
class Test{

public static void main(String[] args){

System.out.println(Adder.add(22,22));

System.out.println(Adder.add(33,33,33));

}}



Experiment 2

class OverloadingCalculation1{

void sum(int a,long b){System.out.println(a+b);}

void sum(int a,int b,int c){System.out.println(a+b+c);}

public static void main(String args[]){

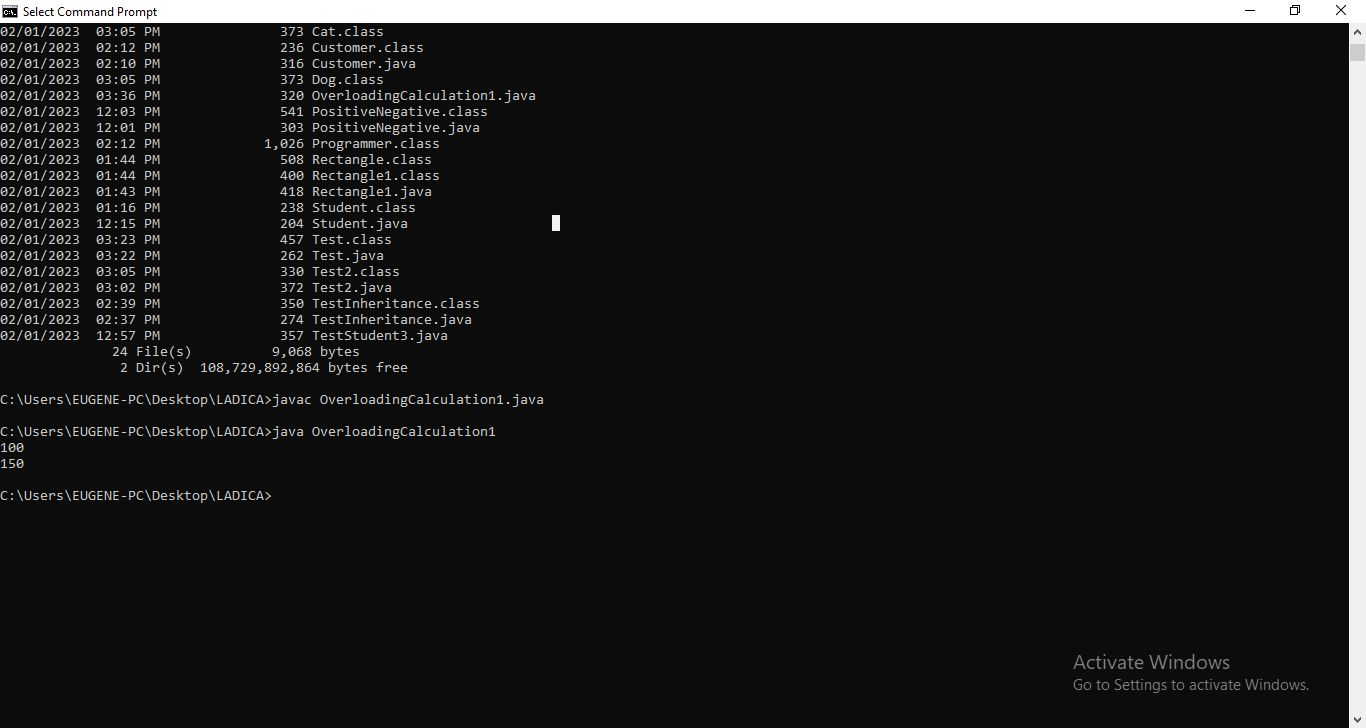
OverloadingCalculation1 obj=new OverloadingCalculation1();

obj.sum(50,50);

obj.sum(50,50,50);

}

}



Experiment 3

class Vehicle{

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

}

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();

obj.run();//calling method

}

}

