**PRACTICAL NO. 1**

**Problem Statement 1 :-**

Write a program to demonstrate method overloading.

Write a program to design an OnlineShoppingPortal. Customers are of two categories, Prime and Regular. “Prime” requires only amount to be paid for checkout and “Regular” customer has to provide amount and promocode. A prime customer gets 30% off if cart value > 1000, else 20% off. A regular customer gets only Rs.10 off if cart value < 500, else 10% off (provided that the code matches “GET10”).

Demonstrate function overloading on a function checkout() which will display the original and discounted amount. Create object in main() to demonstrate function overloading.

**Code:**

**Customer.java**

class Customer {

String category;

ShoppingCart sc;

Customer() {

}

Customer(String category, ShoppingCart sc) {

this.category = category;

this.sc = sc;

}

String getCategory() {

return this.category;

}

ShoppingCart getSc(){

return this.sc;

}

void customer\_display(){

System.out.println("CATEGORY : \t" + this.category);

this.sc.cart\_display();

}

}

**DiscountCalculator.java**

class DiscountCalculator {

float ans=5f;

static float checkout(float amount)

{

float check=1000f;

if(amount>check)

{

float ans=((7\*(amount))/10f);

return ans;

}

else

{

float ans=((8\*(amount))/10f);

return ans;

}

}

static float checkout(float amount, String promocode)

{

float check=500f;

if(amount<check)

{

float ans=amount-10f;

return ans;

}

else if(promocode.equals("GET10"))

{

float ans=((9\*(amount))/10);

return ans;

}

else

{

float ans=amount-10f;

return ans;

}

}

}

**ShoppingCart.java**

class ShoppingCart {

String item\_name;

float item\_amount;

float discounted\_amount;

ShoppingCart() {

}

ShoppingCart(String item\_name, float item\_amount, float discounted\_amount) {

this.item\_name = item\_name;

this.item\_amount = item\_amount;

this.discounted\_amount = discounted\_amount;

}

String getItemName(){

return this.item\_name;

}

float getItemAmount(){

return this.item\_amount;

}

float getDiscountedAmount(){

return this.discounted\_amount;

}

void cart\_display()

{

System.out.println("ITEM NAME : \t" + this.item\_name);

System.out.println("ITEM PRICE : \t" + this.item\_amount);

float d=this.item\_amount-this.discounted\_amount;

System.out.println("DISCOUNT : \t" + d);

System.out.println("NET PRICE : \t" + this.discounted\_amount);

System.out.println();

}

}

**Test.java**

class Test {

public static void main(String[] args) {

ShoppingCart s\_cart1 = new ShoppingCart();

ShoppingCart s\_cart2 = new ShoppingCart();

ShoppingCart s\_cart3 = new ShoppingCart();

ShoppingCart s\_cart4 = new ShoppingCart();

ShoppingCart s\_cart5 = new ShoppingCart();

Customer cus\_1 = new Customer("PRIME", s\_cart1);

Customer cus\_2 = new Customer("PRIME", s\_cart2);

Customer cus\_3 = new Customer("REGULAR", s\_cart3);

Customer cus\_4 = new Customer("REGULAR", s\_cart4);

Customer cus\_5 = new Customer("REGULAR", s\_cart5);

///////////////////////////// PRIME GREATER THAN 1000//////////////////////////////////////////////////////

System.out.println();

System.out.println("PRIME CUSTOMER WITH GREATER THAN 1000 AMOUNT");

s\_cart1.item\_name = "Mobile Phone";

s\_cart1.item\_amount = 1500f;

float dis1;

String cus\_type1 = cus\_1.getCategory();

if (cus\_type1.equals("REGULAR")) {

dis1 = DiscountCalculator.checkout(cus\_1.sc.item\_amount, "GET10");

} else {

dis1 = DiscountCalculator.checkout(cus\_1.sc.item\_amount);

}

s\_cart1.discounted\_amount = dis1;

System.out.println();

cus\_1.customer\_display();

System.out.println();

//////////////////////////////// PRIME LESS THAN 1000///////////////////////////////////////////////////

System.out.println("PRIME CUSTOMER WITH LESS THAN 1000 AMOUNT");

s\_cart2.item\_name = "Bike";

s\_cart2.item\_amount = 500f;

float dis2;

String cus\_type2 = cus\_2.getCategory();

if (cus\_type2.equals("REGULAR")) {

dis2 = DiscountCalculator.checkout(cus\_2.sc.item\_amount, "GET10");

} else {

dis2 = DiscountCalculator.checkout(cus\_2.sc.item\_amount);

}

s\_cart2.discounted\_amount = dis2;

System.out.println();

cus\_2.customer\_display();

System.out.println();

///////////////////////////// REGULAR GREATER THAN 500 CORRECT CODE//////////////////////////////////////////////////////

System.out.println("REGULAR CUSTOMER WITH GREATER THAN 500 AMOUNT & CORRECT CODE");

s\_cart3.item\_name = "CAR";

s\_cart3.item\_amount = 1000f;

float dis3;

String cus\_type3 = cus\_3.getCategory();

if (cus\_type3.equals("REGULAR")) {

dis3 = DiscountCalculator.checkout(cus\_3.sc.item\_amount, "GET10");

} else {

dis3 = DiscountCalculator.checkout(cus\_3.sc.item\_amount);

}

s\_cart3.discounted\_amount = dis3;

System.out.println();

cus\_3.customer\_display();

System.out.println();

///////////////////////////// REGULAR GREATER THAN 500 INCORRECT CODE//////////////////////////////////////////////////////

System.out.println("REGULAR CUSTOMER WITH GREATER THAN 500 AMOUNT & INCORRECT CODE");

s\_cart4.item\_name = "CAR";

s\_cart4.item\_amount = 1000f;

float dis4;

String cus\_type4 = cus\_4.getCategory();

if (cus\_type4.equals("REGULAR")) {

dis4 = DiscountCalculator.checkout(cus\_4.sc.item\_amount, "GET50");

} else {

dis4 = DiscountCalculator.checkout(cus\_4.sc.item\_amount);

}

s\_cart4.discounted\_amount = dis4;

System.out.println();

cus\_4.customer\_display();

System.out.println();

///////////////////////////// REGULAR LESS THAN 500//////////////////////////////////////////////////////

System.out.println("REGULAR CUSTOMER WITH LESS THAN 500 AMOUNT ");

s\_cart5.item\_name = "PEN";

s\_cart5.item\_amount = 80f;

float dis5;

String cus\_type5 = cus\_5.getCategory();

if (cus\_type5.equals("REGULAR")) {

dis5 = DiscountCalculator.checkout(cus\_5.sc.item\_amount, "GET10");

} else {

dis5 = DiscountCalculator.checkout(cus\_5.sc.item\_amount);

}

s\_cart5.discounted\_amount = dis5;

System.out.println();

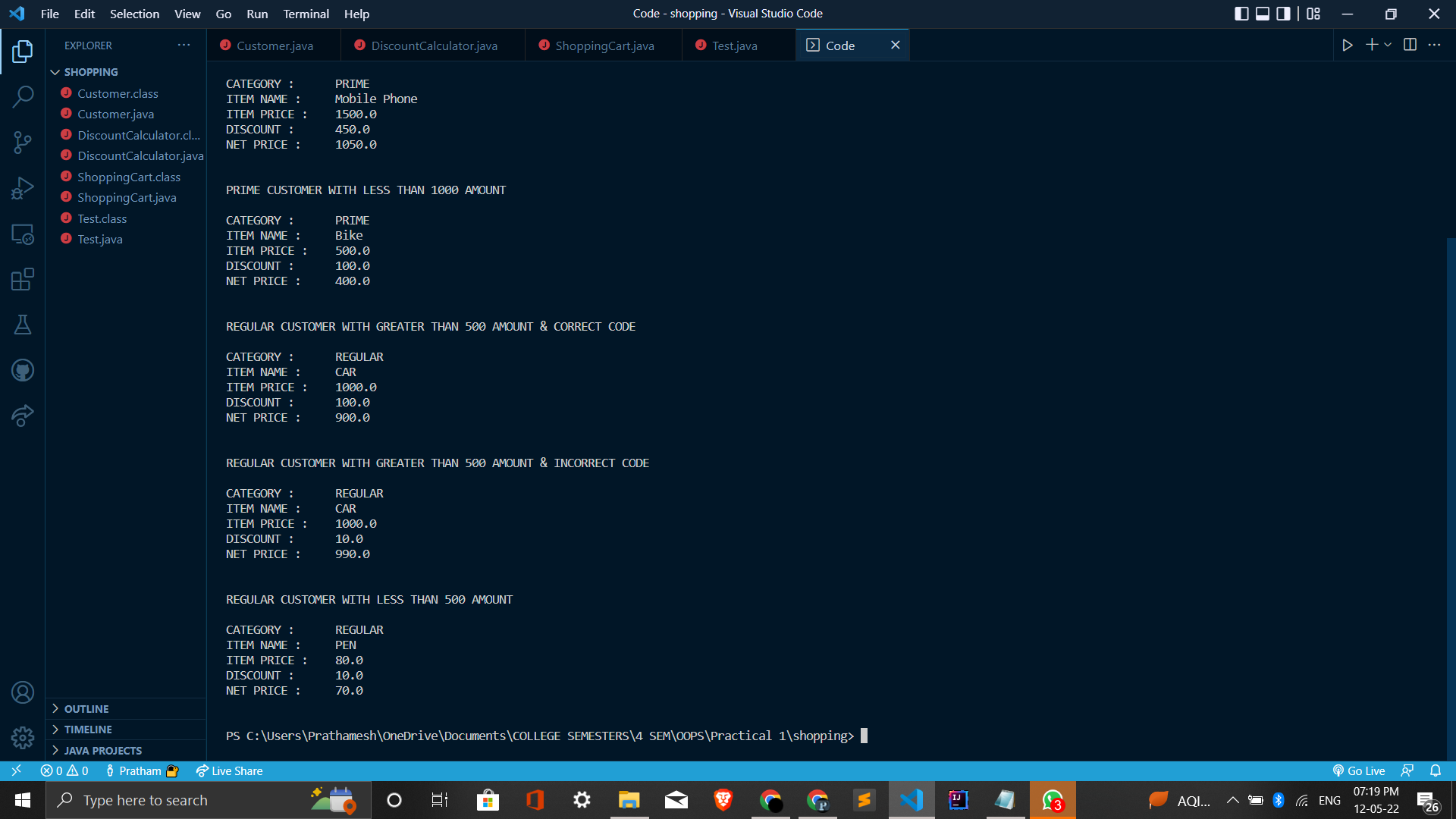
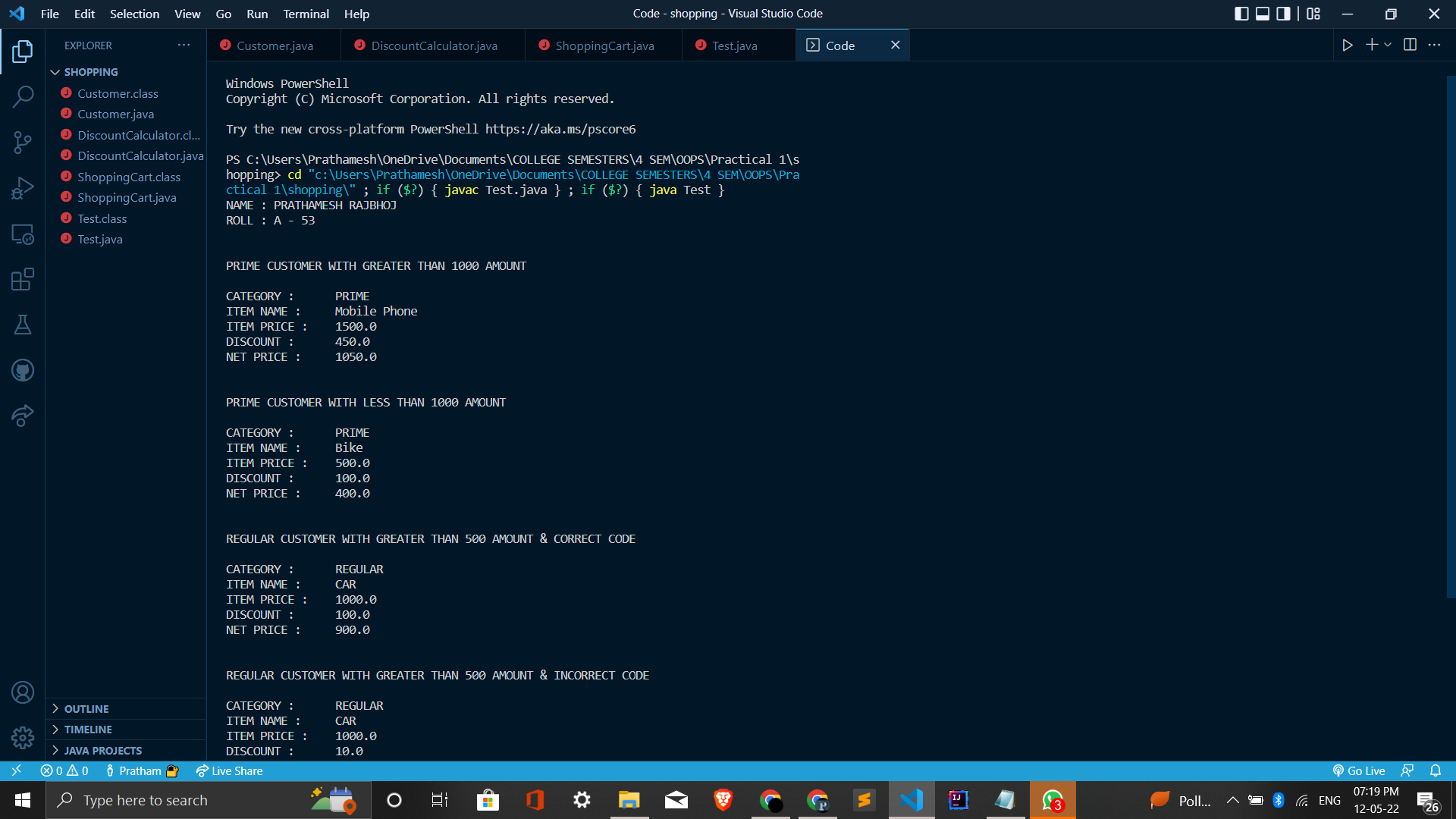
cus\_5.customer\_display();

System.out.println();

}

}

**OUTPUT :**

****

**Problem Statement 2 :-**

Create a class complex and provide functionalities to add and multiply 2 complex numbers. Class Complex will have real and imaginary as the data

members. Write appropriate main() to demonstrate the functionalities.

**Code:**

**Complex.java**

class Complex {

int real;

int img;

Complex() {

}

Complex(int real, int img) {

this.real = real;

this.img = img;

}

void display() {

System.out.println("Real Part : "+this.real + "\t\tImaginary Part : " + this.img + "\n");

}

int getReal() {

return this.real;

}

int getImg() {

return this.img;

}

void setReal(int real) {

this.real = real;

}

void setImg(int img) {

this.img = img;

}

// The formula for adding two complex numbers is:

// (a + ib) + (c + id) = (a+c) + i(b+d)

static Complex addComplex(Complex x, Complex y) {

Complex res = new Complex();

res.real = x.real + y.real;

res.img = x.img + y.img;

return res;

}

// The formula for multiplying two complex numbers is:

// (a + ib) \* (c + id) = (ac -bd) + i(ad + bc)

static Complex multiplyComplex(Complex x, Complex y) {

Complex res = new Complex();

int a = x.real;

int b = x.img;

int c = y.real;

int d = y.img;

res.real = (a \* c - b \* d);

res.img = (a \* d + b \* c);

return res;

}

}

**Test.java**

public class Test {

public static void main(String[] args) {

Complex x = new Complex(2,3);

Complex y = new Complex(7,10);

System.out.println("NAME : PRATHAMESH RAJBHOJ");

System.out.println("ROLL : A - 53");

System.out.println();

x.display();

y.display();

Complex a = Complex.addComplex(x,y);

Complex b = Complex.multiplyComplex(x,y);

System.out.println("RESULT OF ADDITION :\n");

a.display();

System.out.println("RESULT OF MULTIPLICATION :\n");

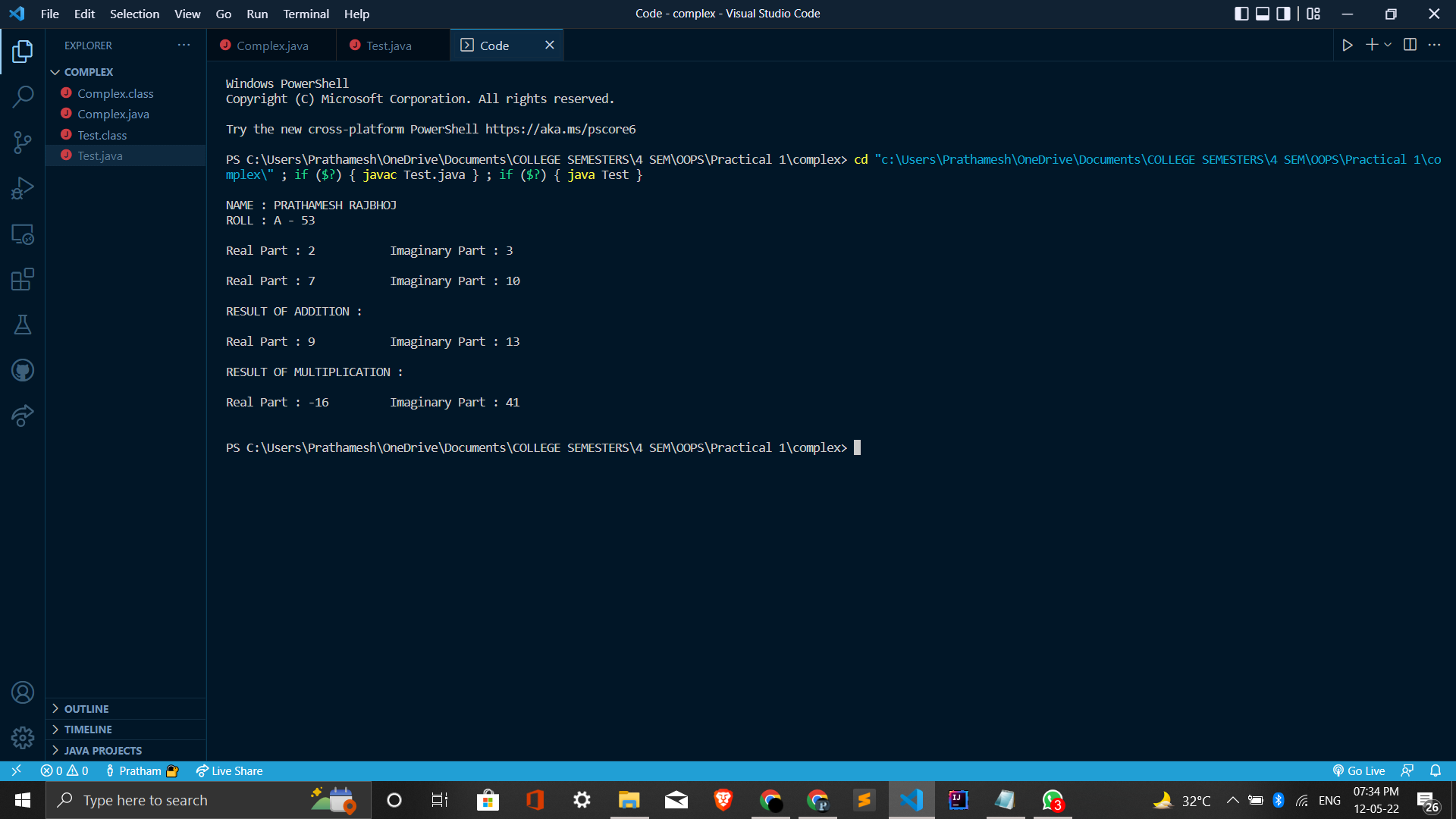
b.display();

System.out.println();

}

}

**OUTPUT :**

****