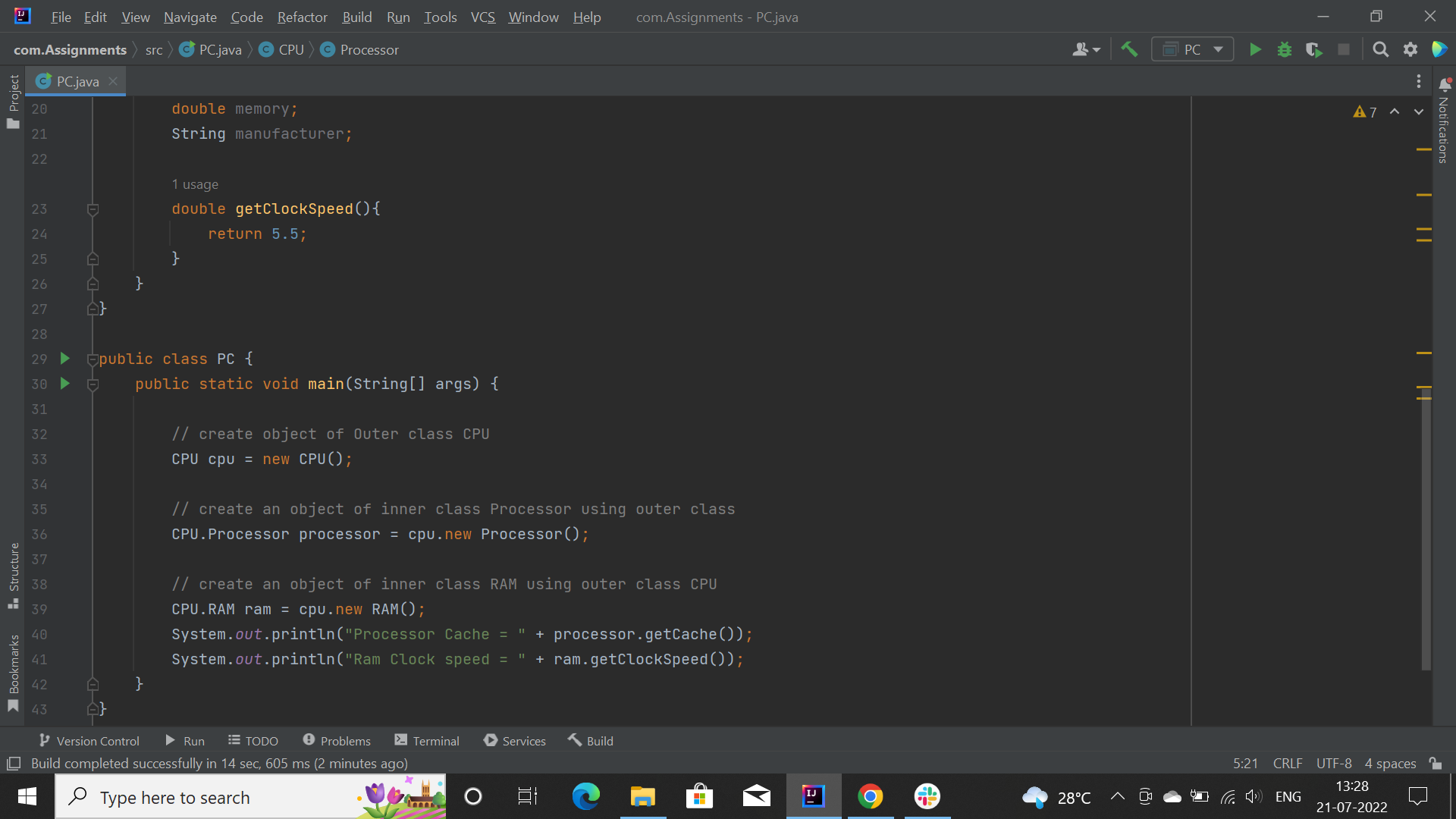
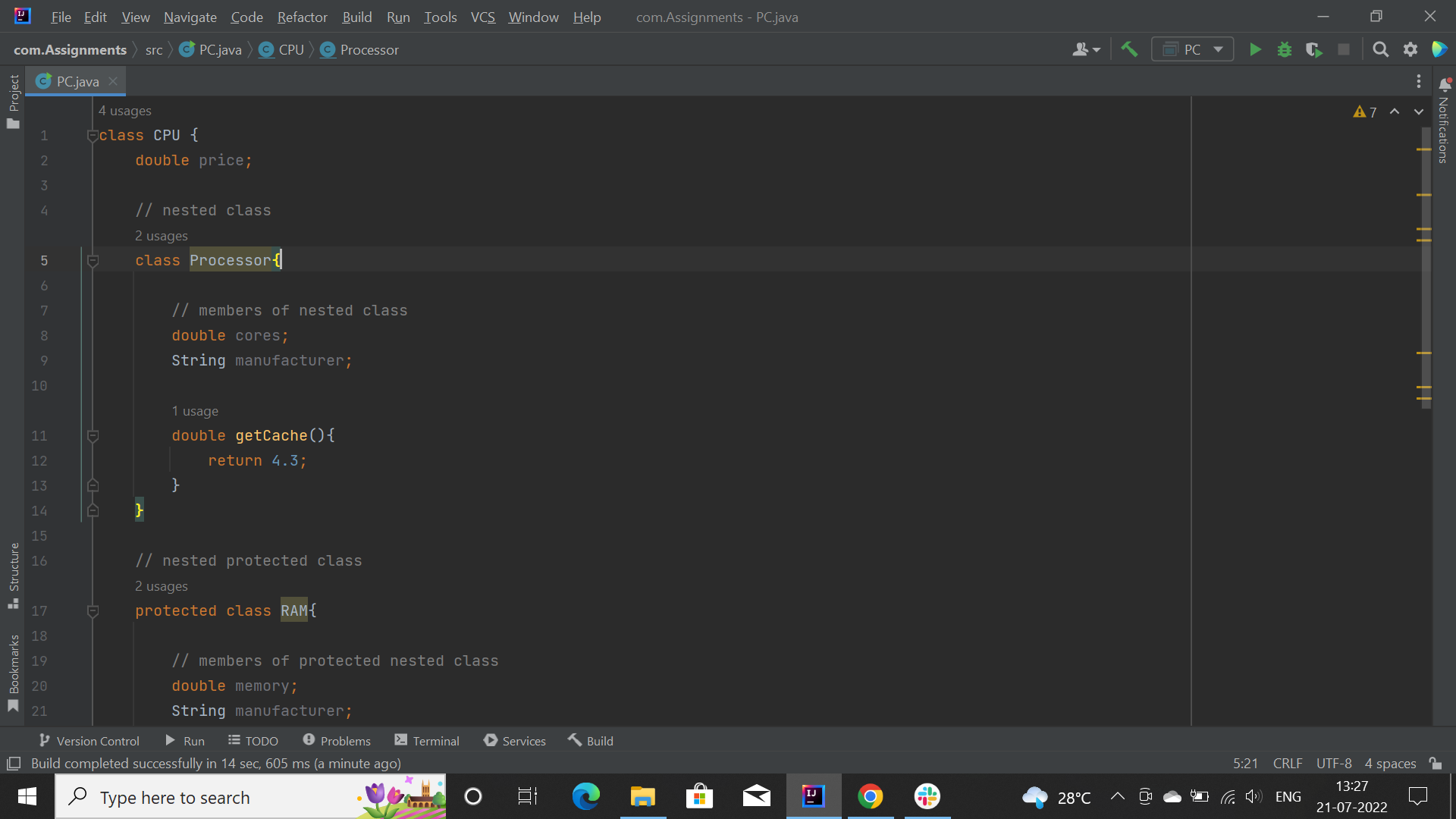
**JAVA ASSIGNMENT (DAY-3)**

1. **Complete the below code, by writing main() in different class.**

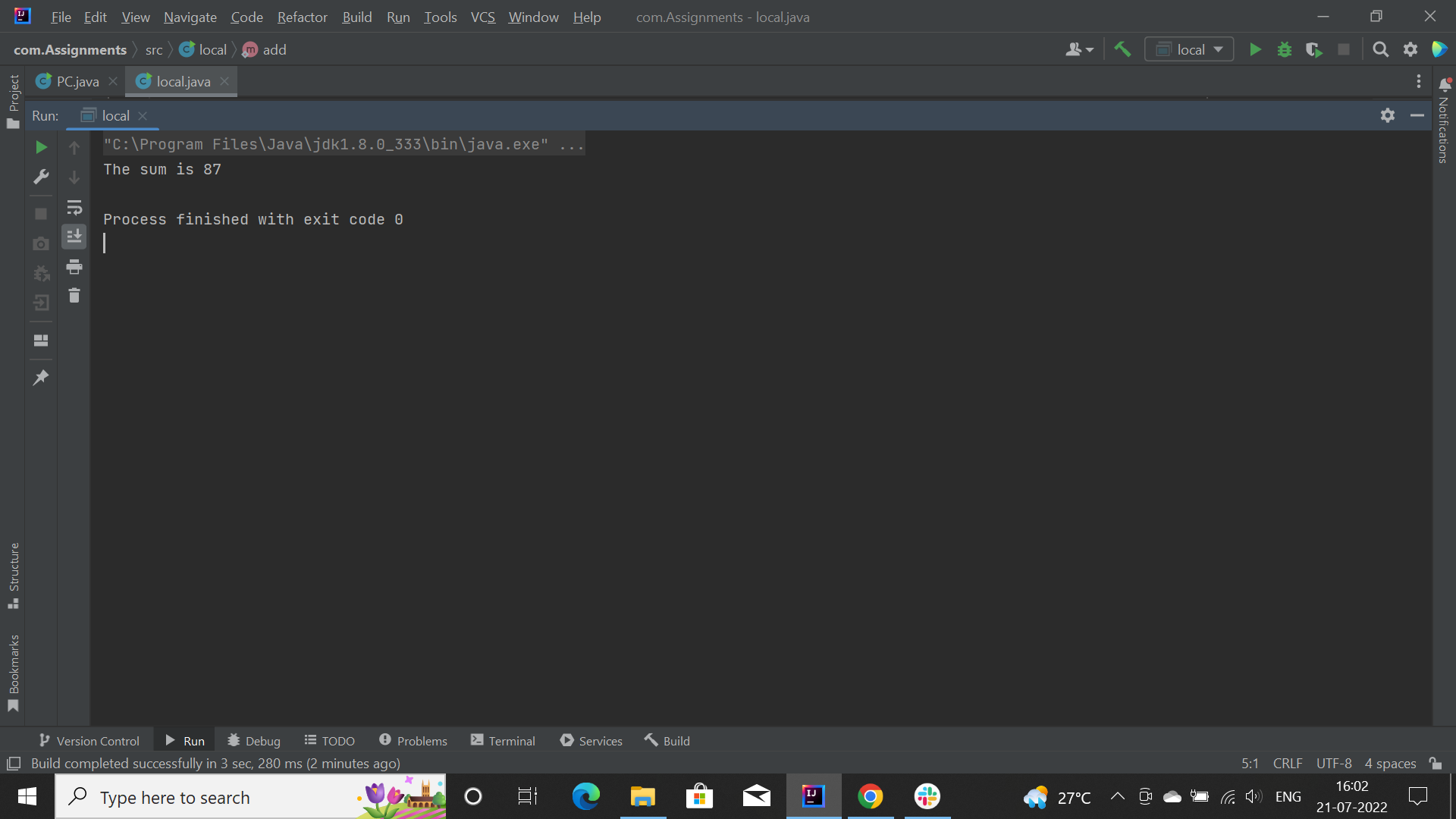
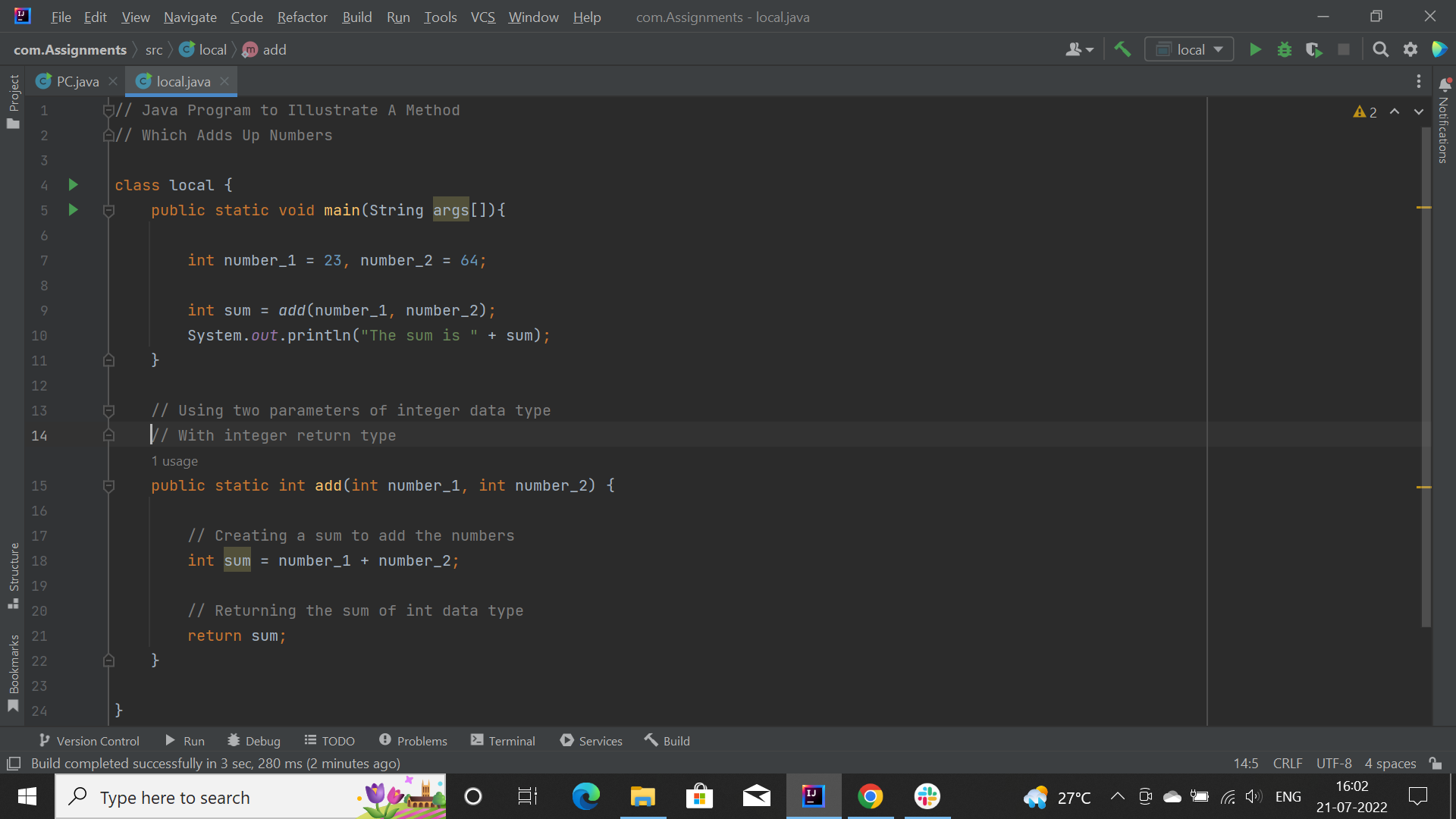
****

In the above program, there are two nested classes: RAM and inside the outer class: CPU. We can declare the inner class as protected. Hence, we have declared the RAM class as protected.

Inside the Main class, we first created an instance of an outer class CPU named cpu.

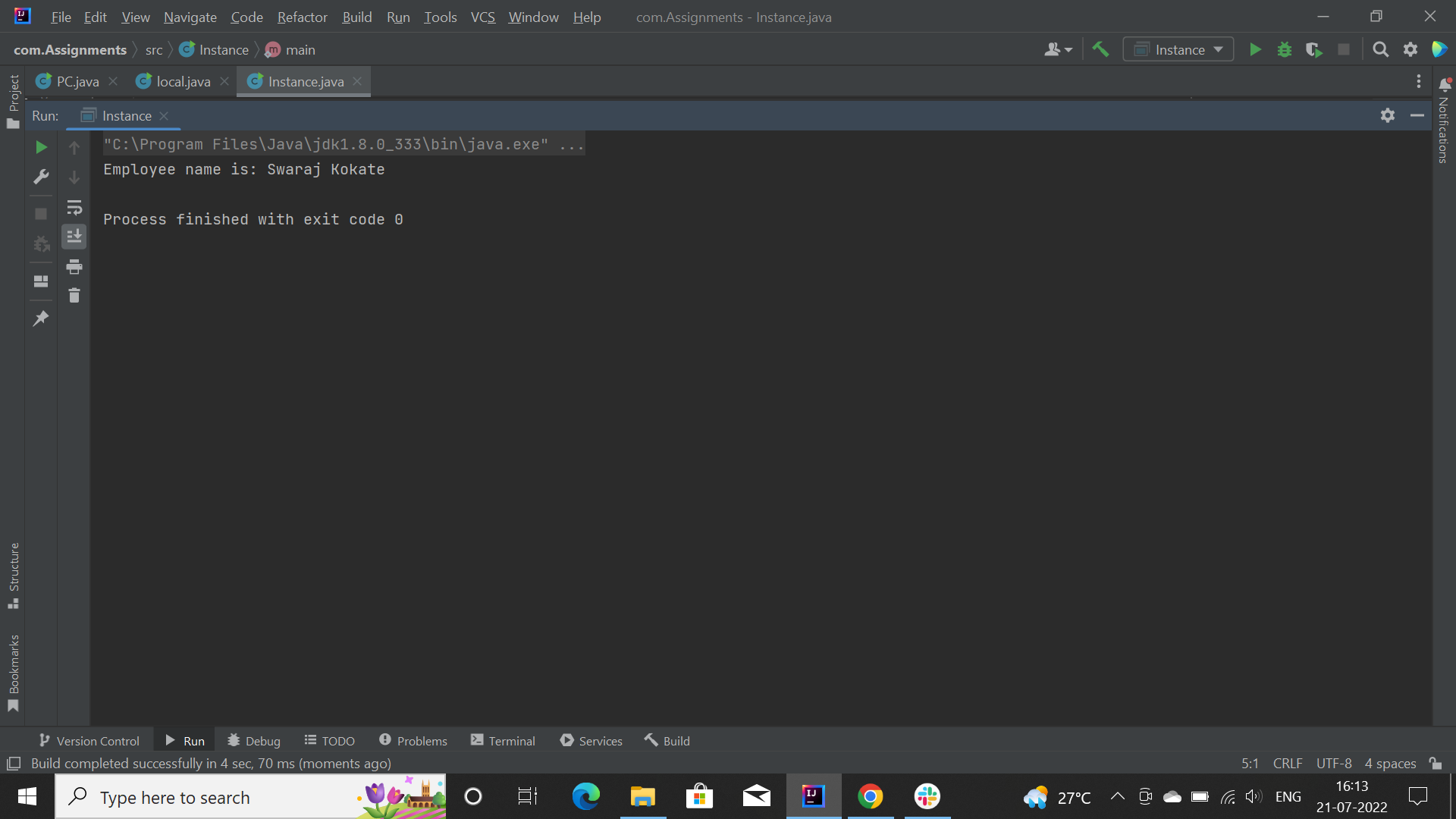
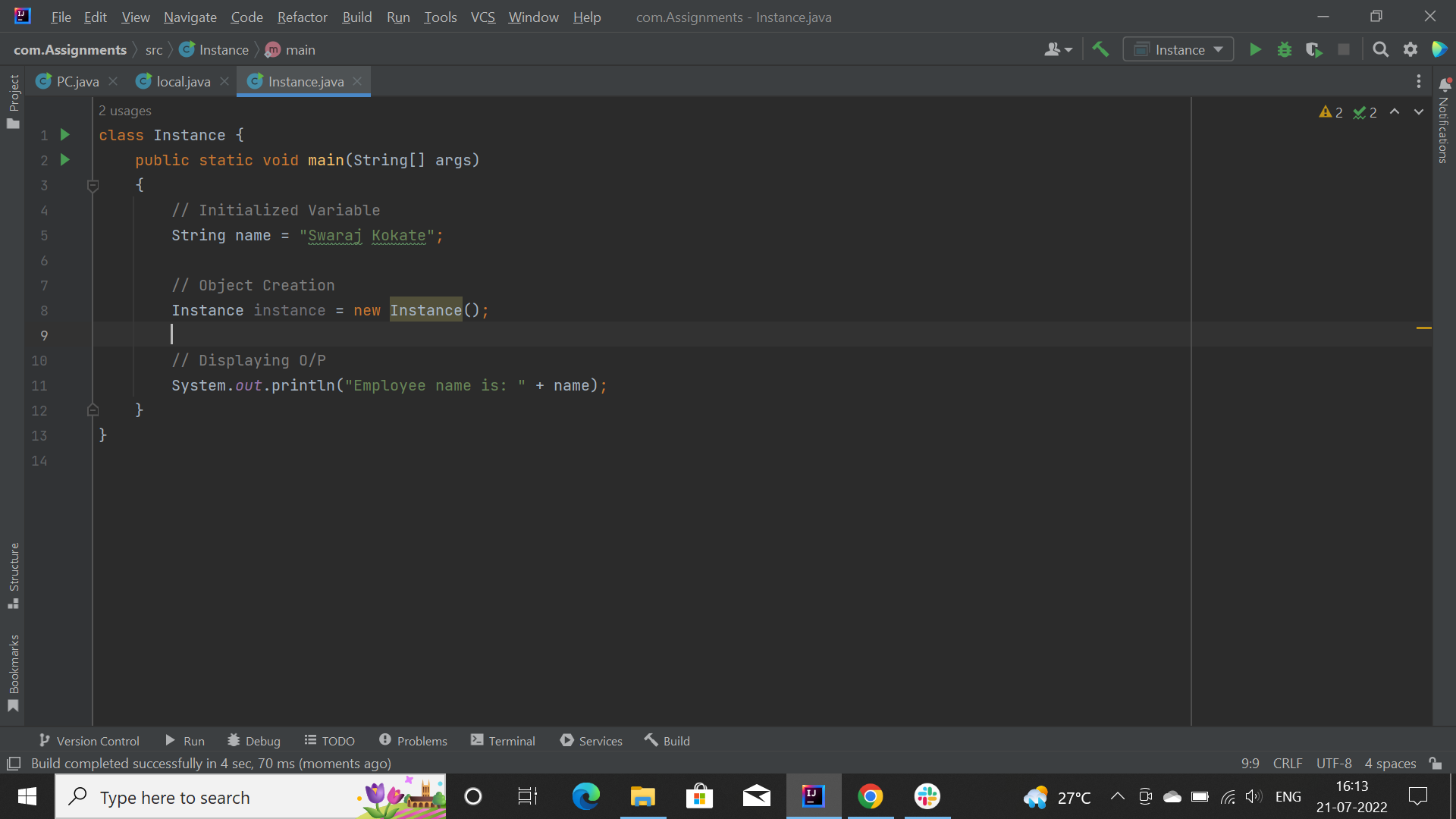
We can use (.) operator to create instance of inner class.

1. **Write a program of local, instance and static variable.**
2. **Local**



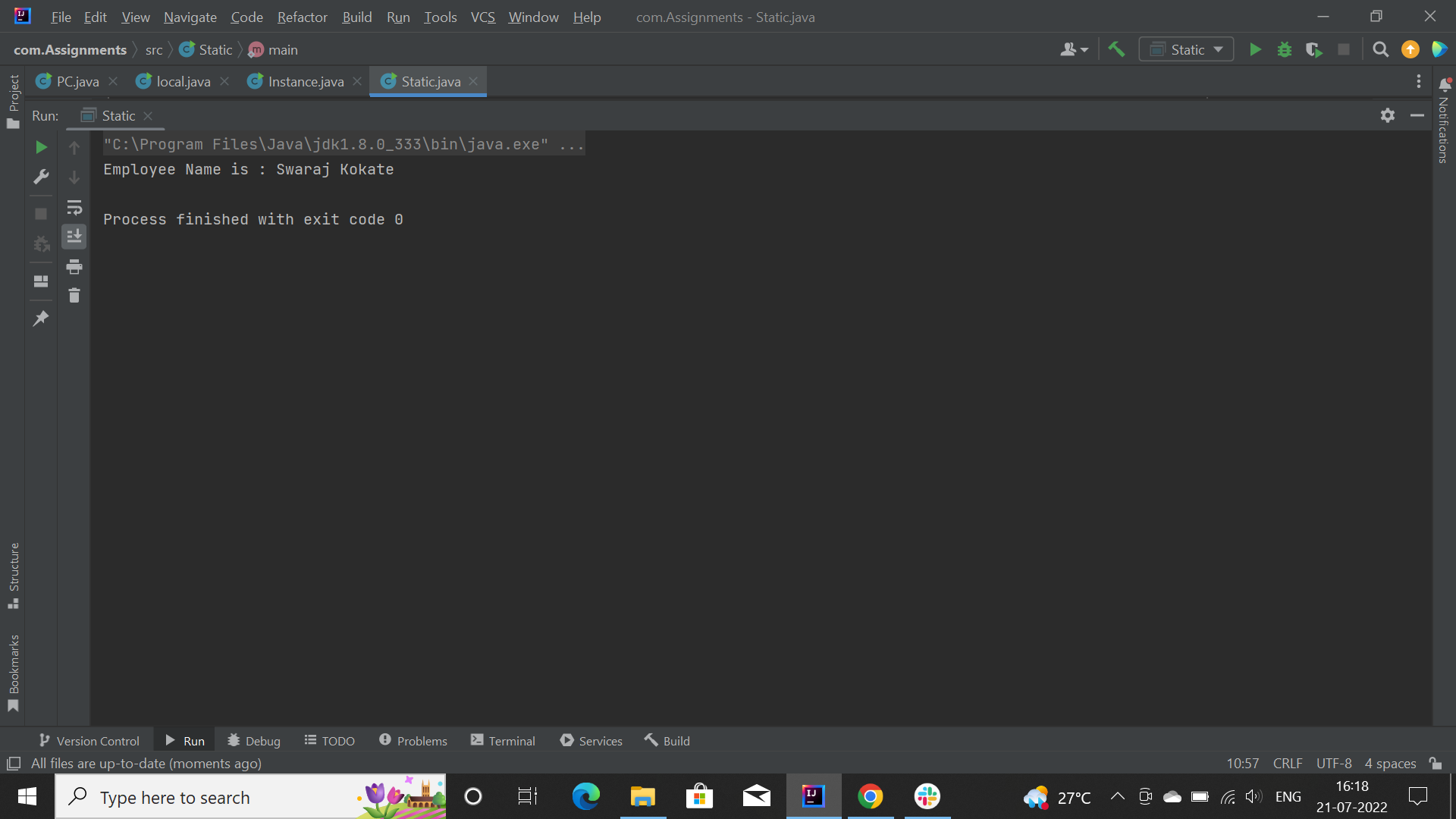
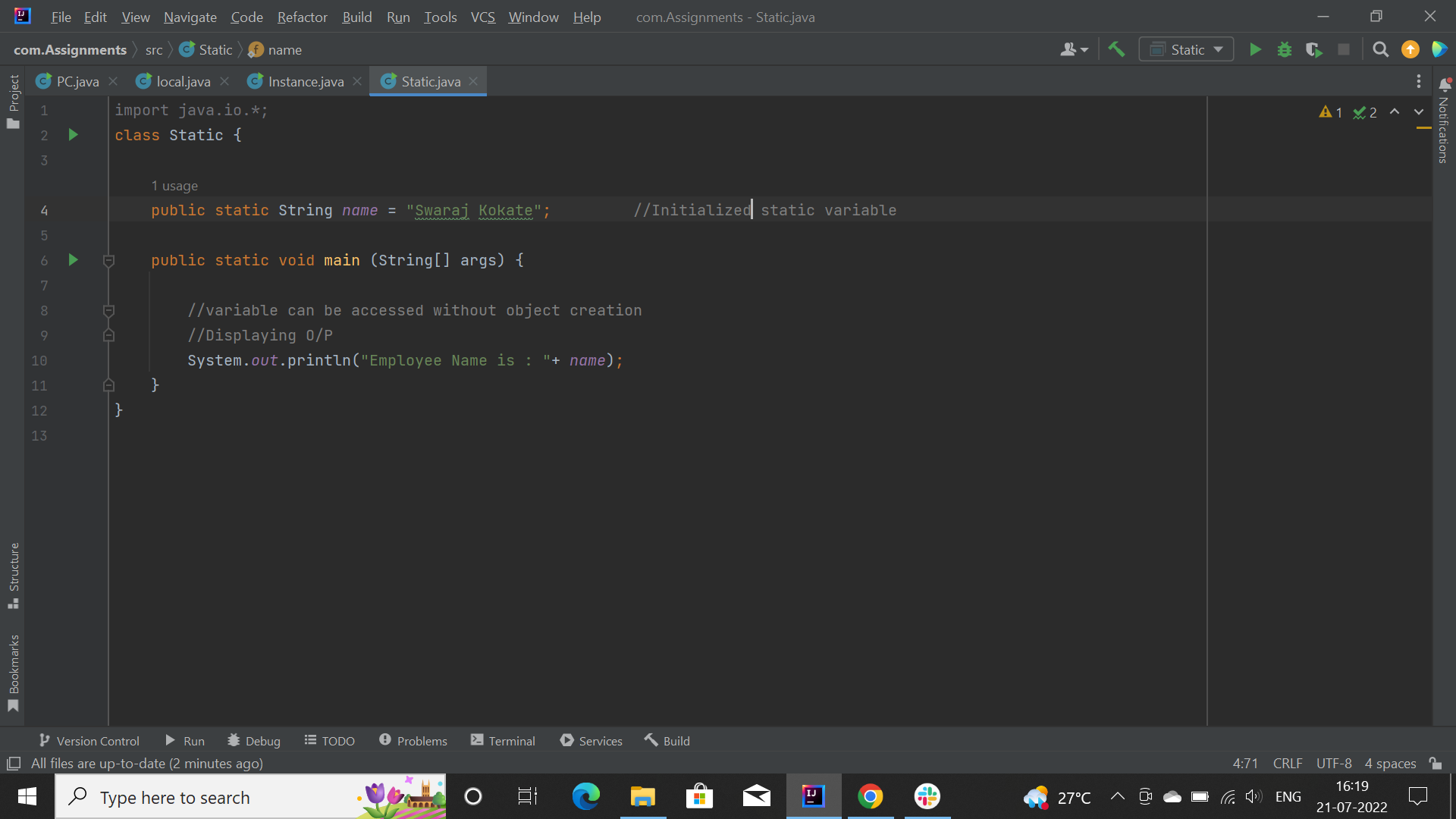
Here, we have three variables number\_1, number\_2 and sum. The scope of this variables is within their respective function. Any attempt to access this variables outside the function, will throw an error.

1. **Instance:**

****

Here, I have created instance variable named ‘instance’ of class ‘Instance’

**3) Static:**

****

Here, I have created static variable ‘name’.

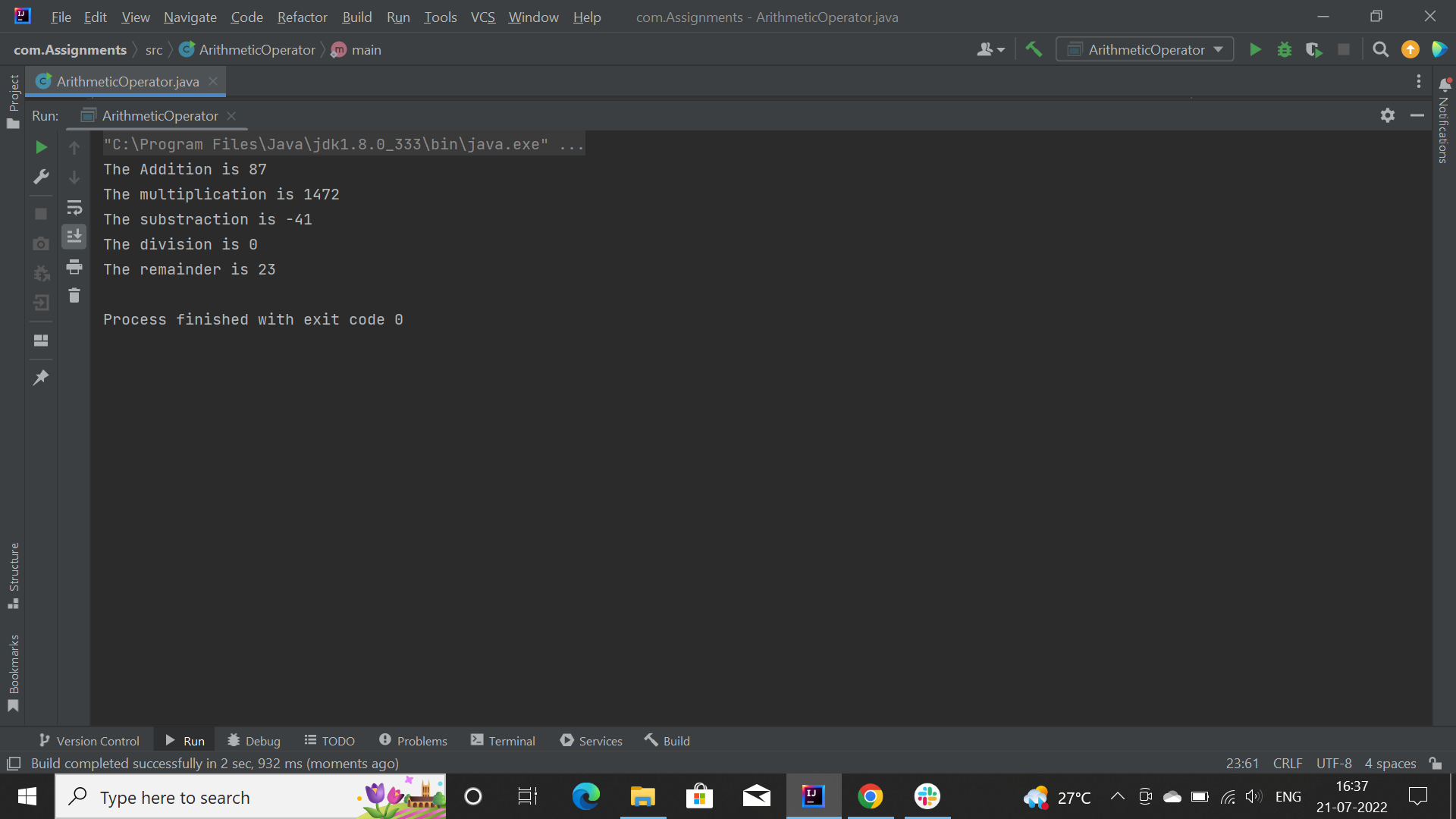
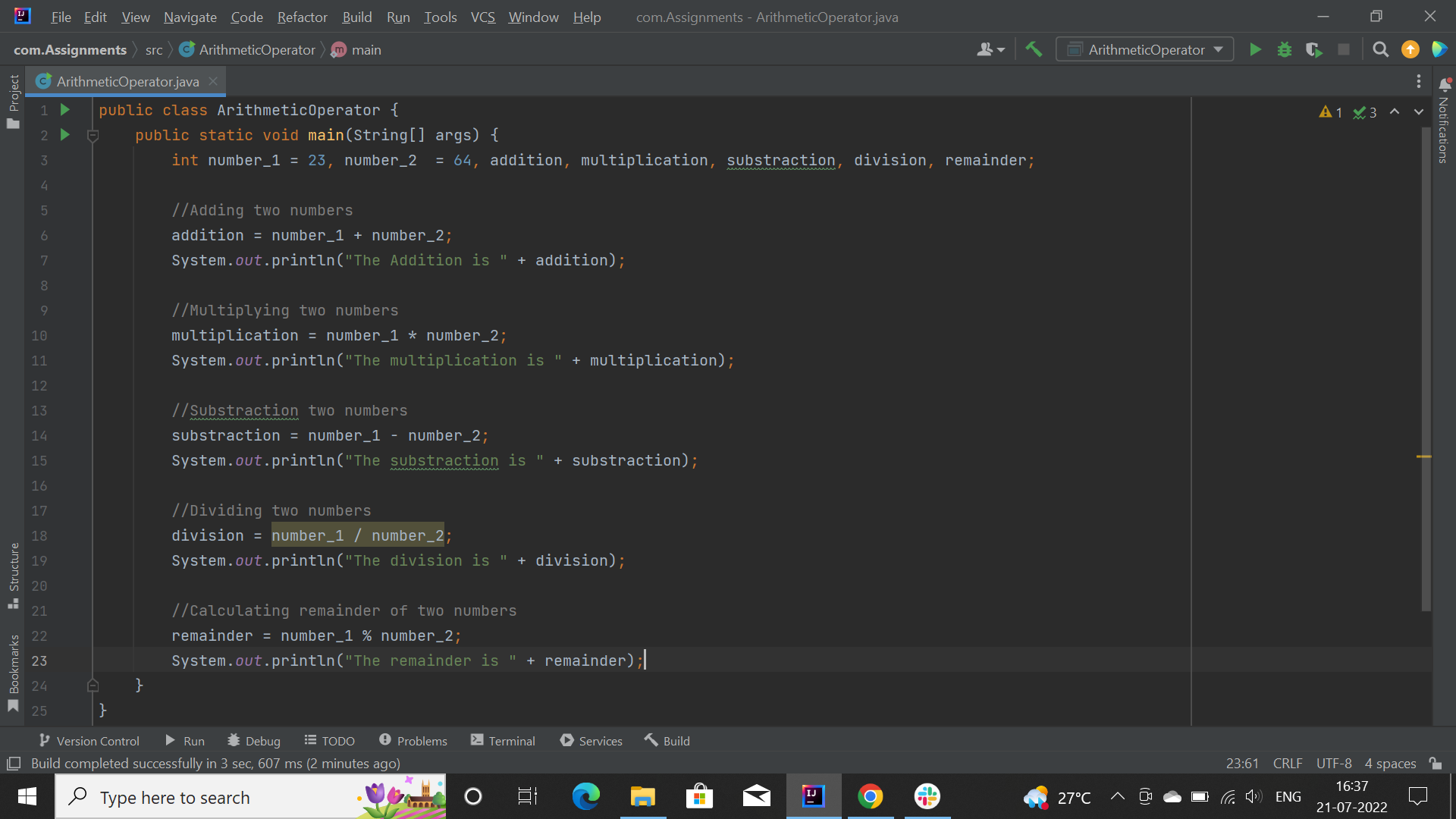
**3) Write a program of operators. Explain the concept of operators and check the difference between logical and bitwise operator.**

Operators are special symbols that perform specific operations on one, two, or three *operands*, and then return a result.

**Types:**

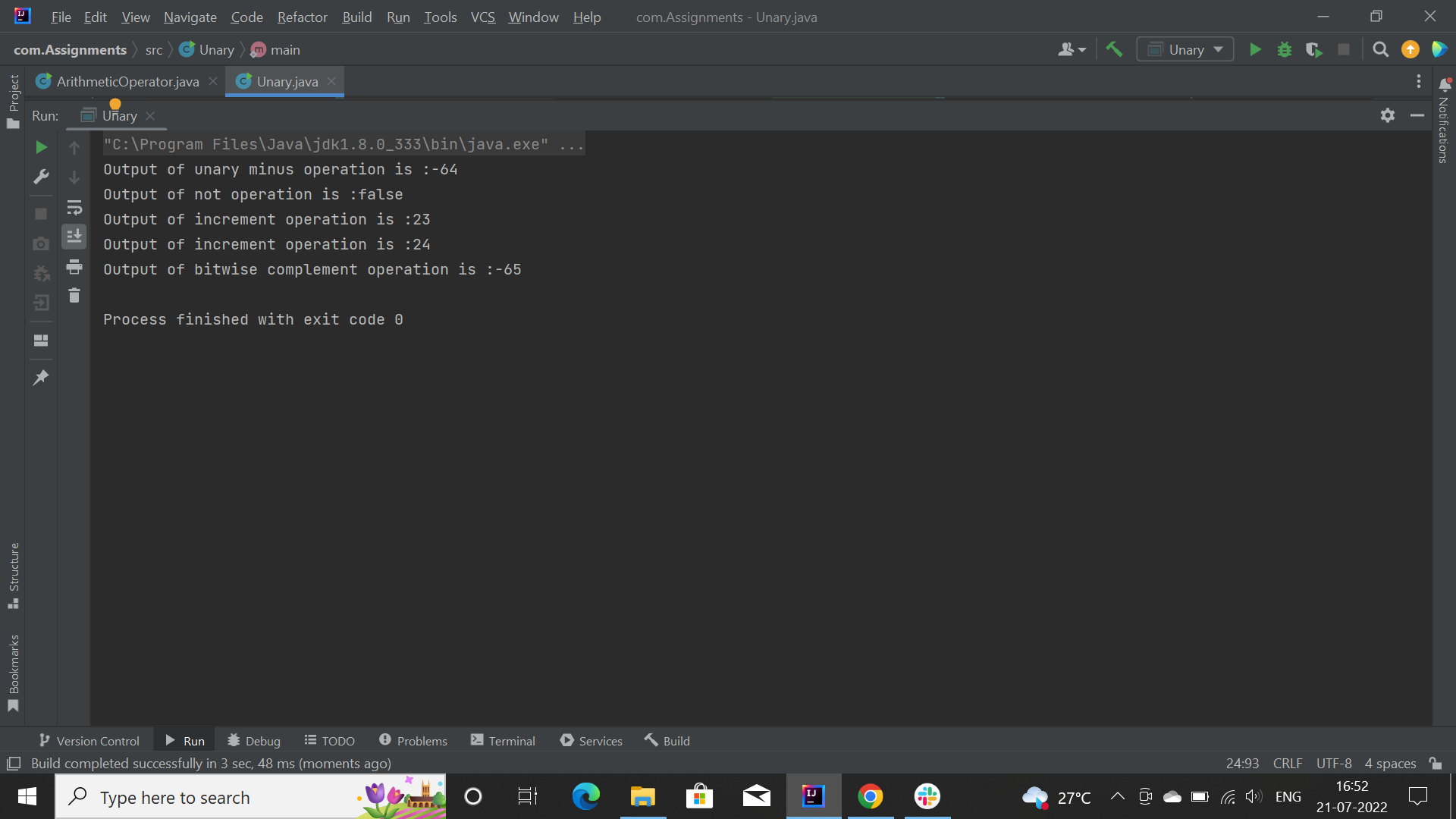
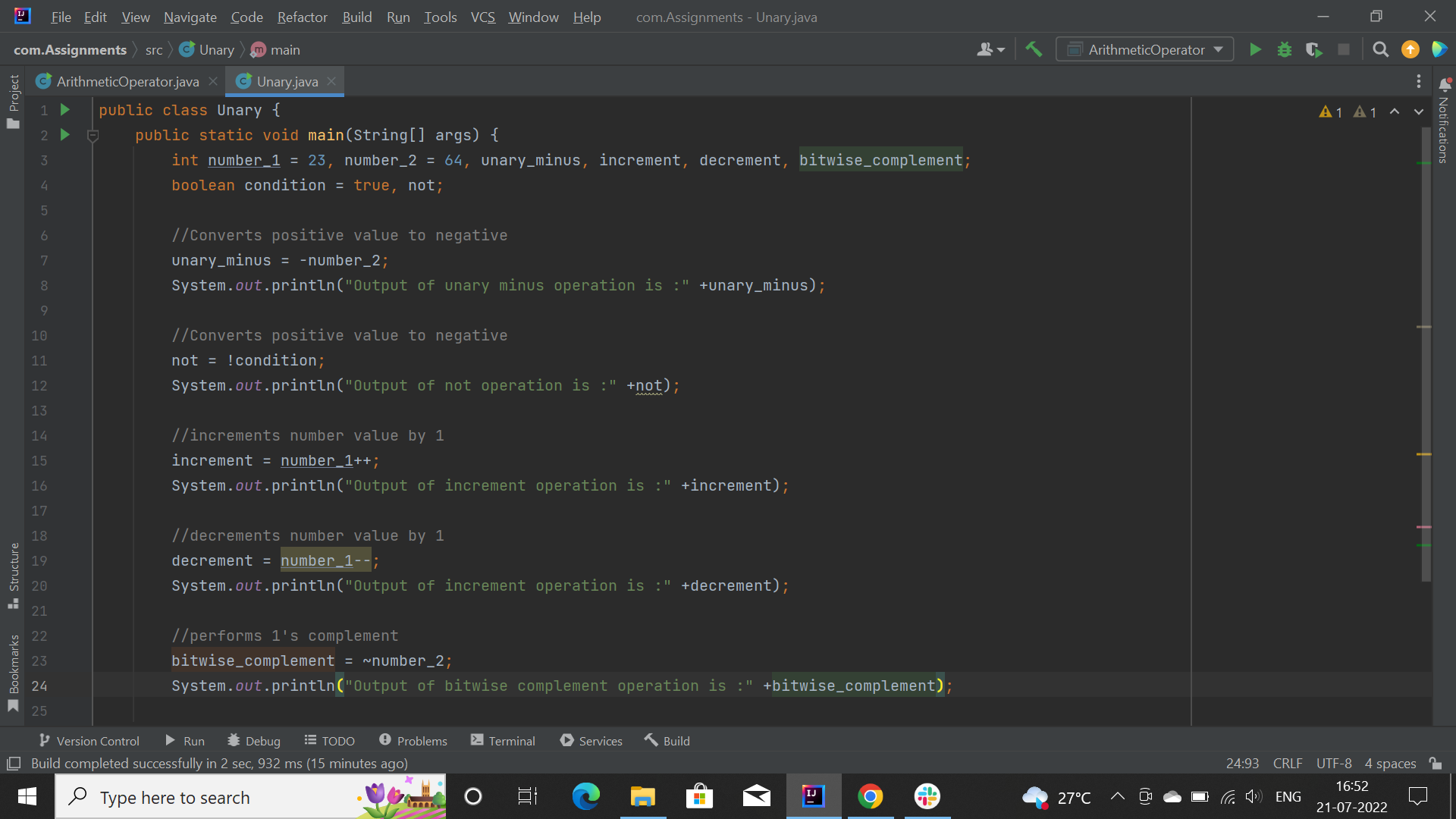
1. **Arithmetic operators:**

These operators involve the mathematical operators that can be used to perform various simple or advanced arithmetic operations on the primitive data types referred to as the operands. These operators consist of various unary and binary operators that can be applied on a single or two operands.

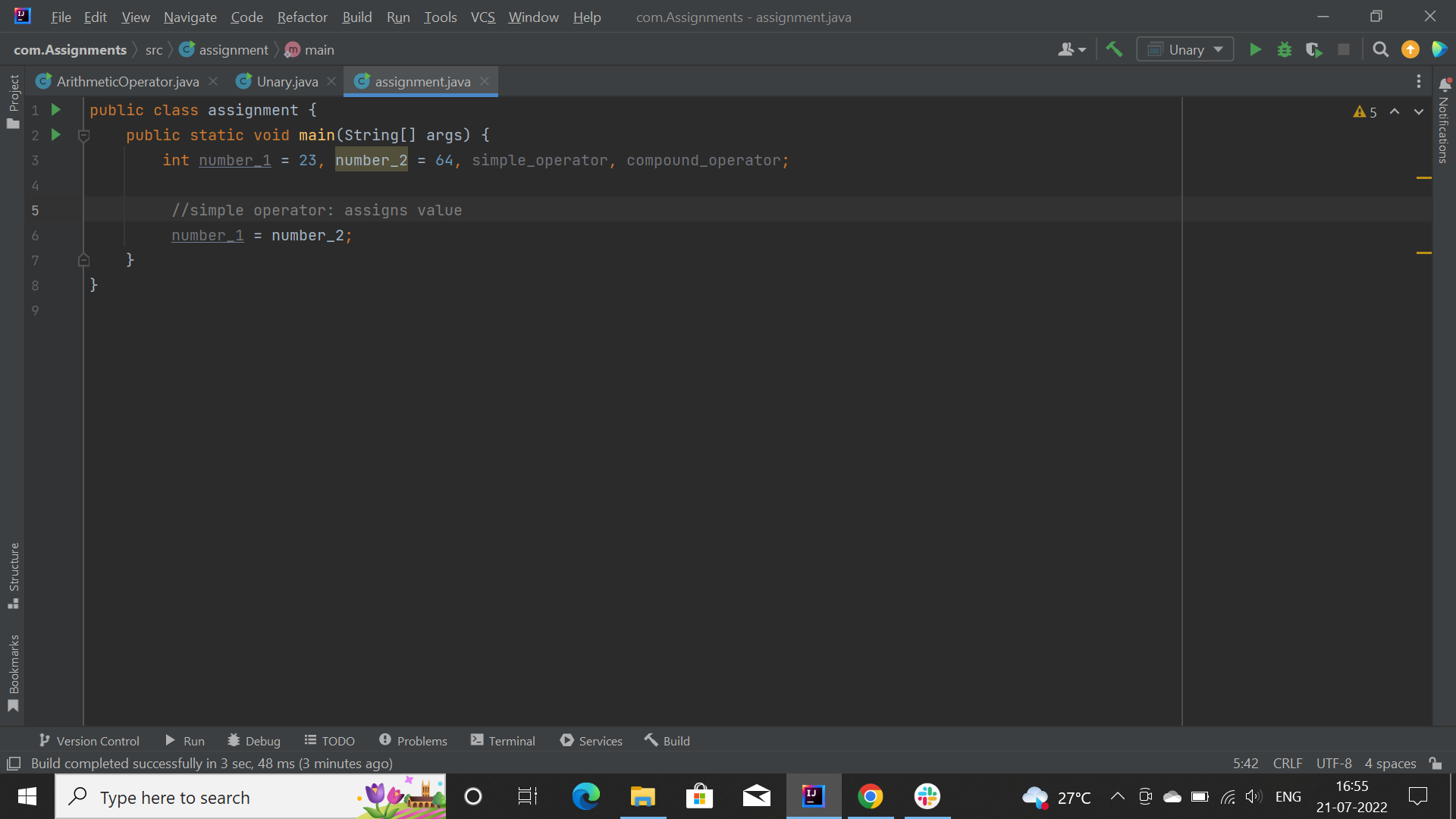


1. **Unary operators:**

Java unary operators are the types that need only one operand to perform any operation like increment, decrement, negation, etc.

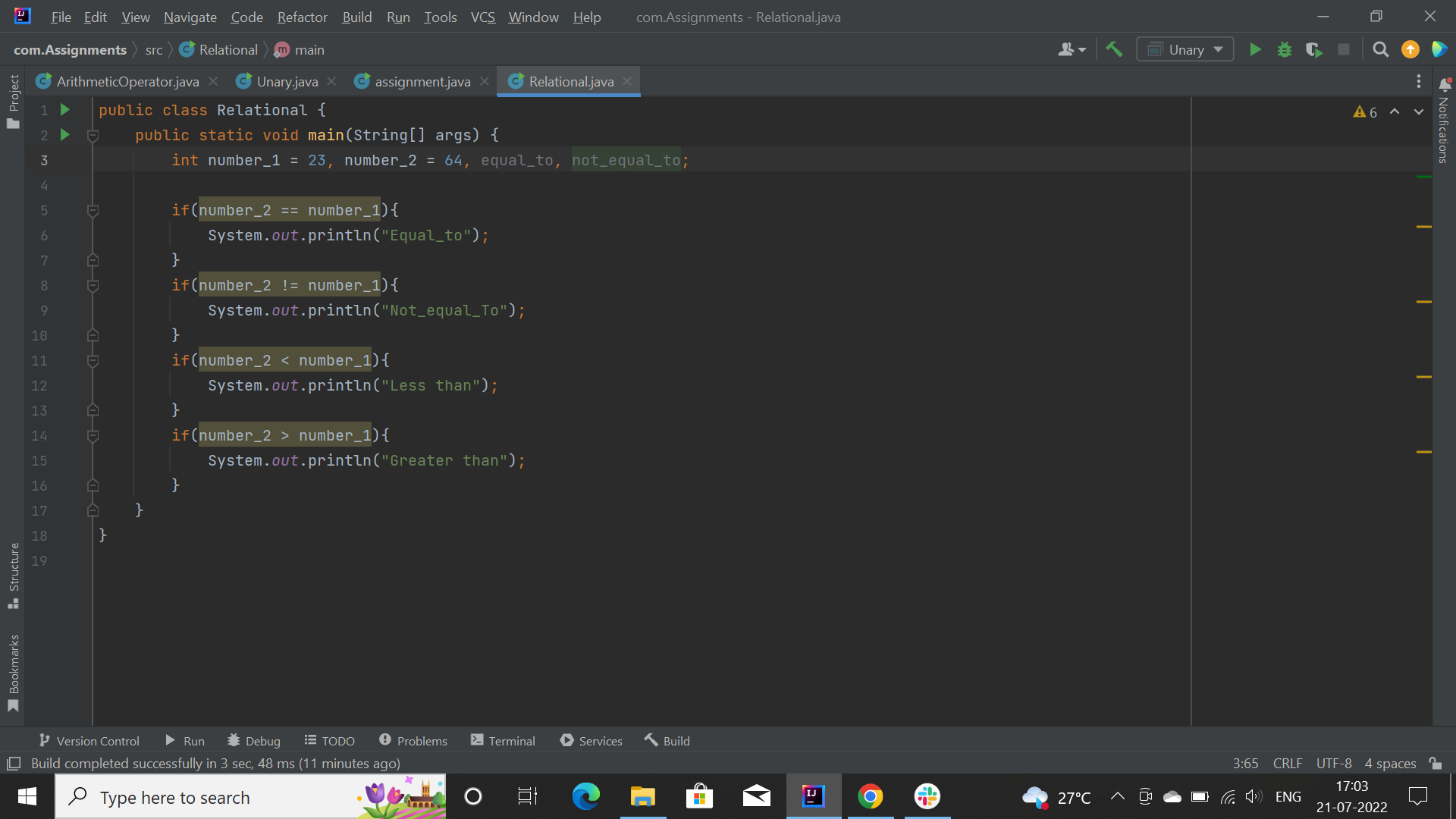


1. **Assignment operator**

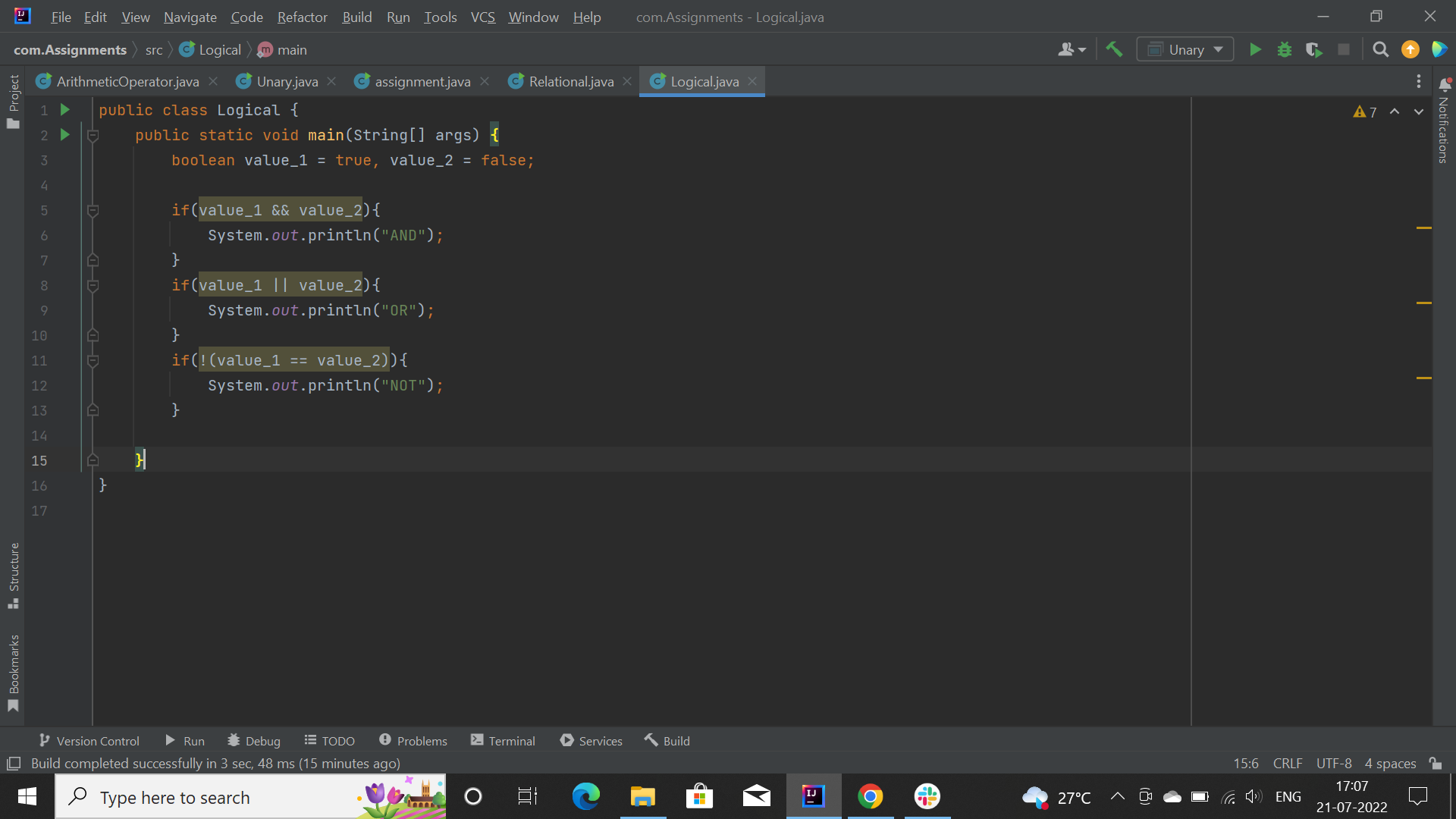
****

**2. Compound Assignment Operator:** The Compound Operator is used where +,-,\*, and / is used along with the = operator.

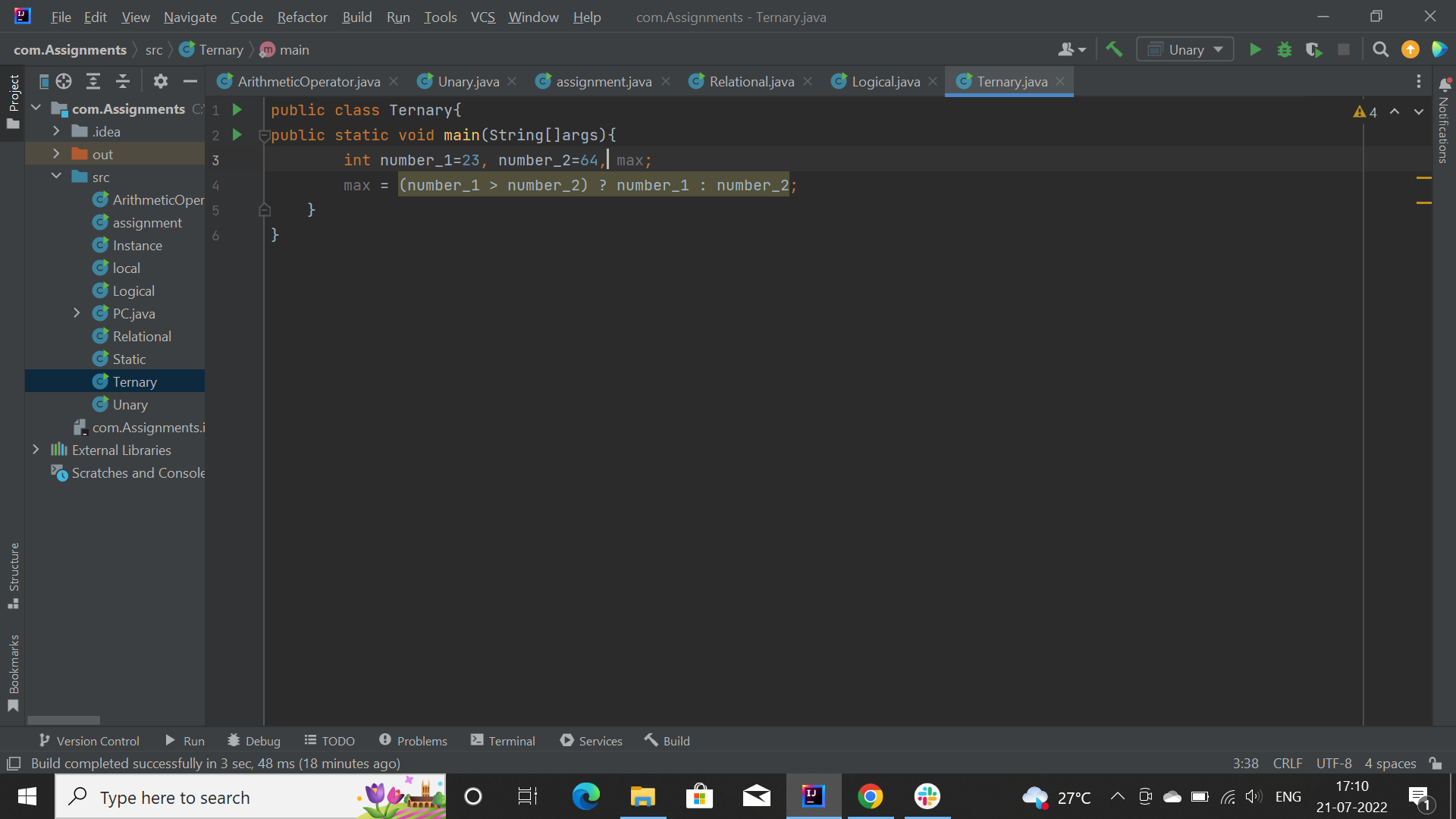
1. **Relational operators:**

****

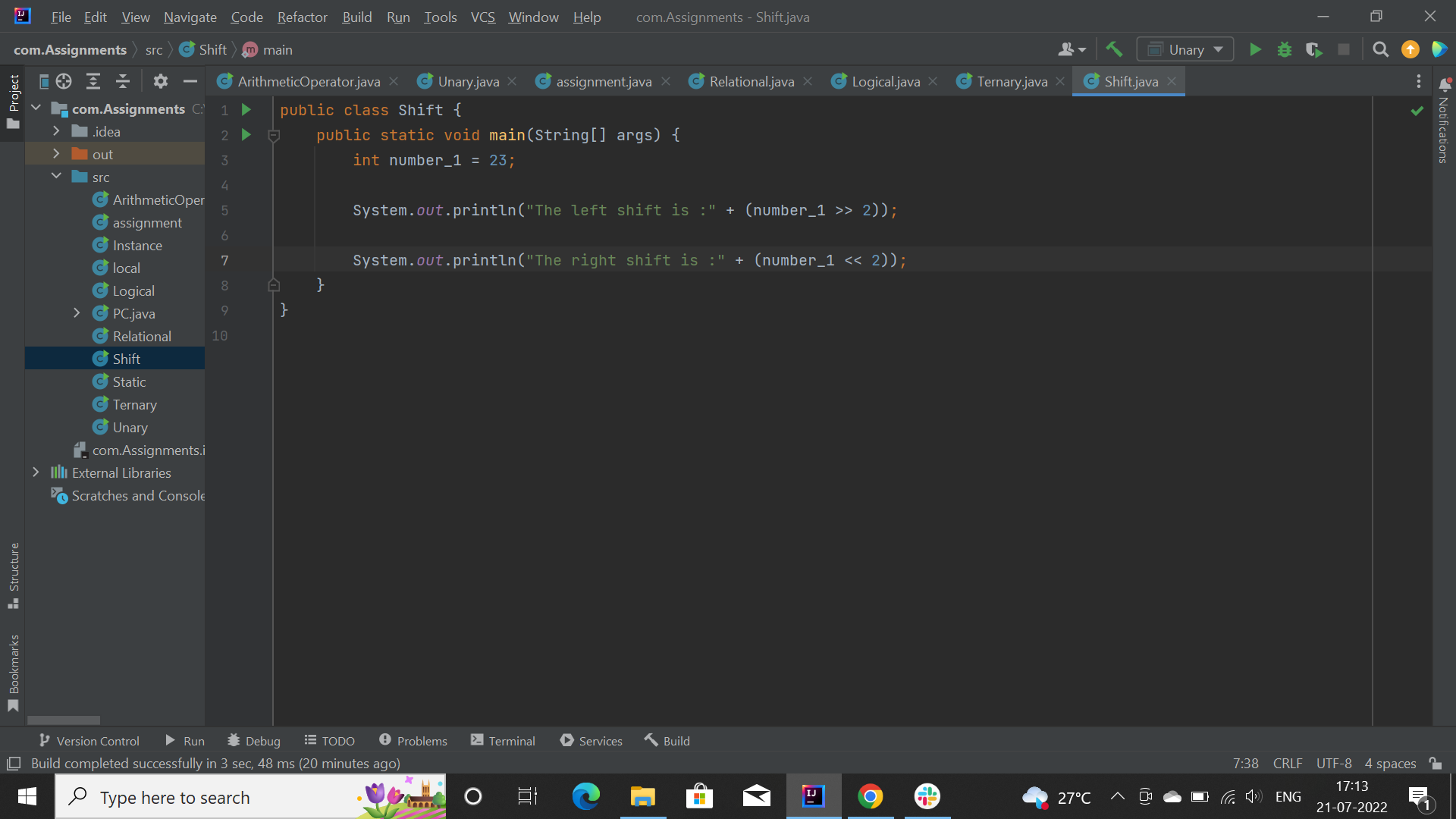
1. **Logical operator:**

****

1. **Ternary:**

****

1. **Shift:**

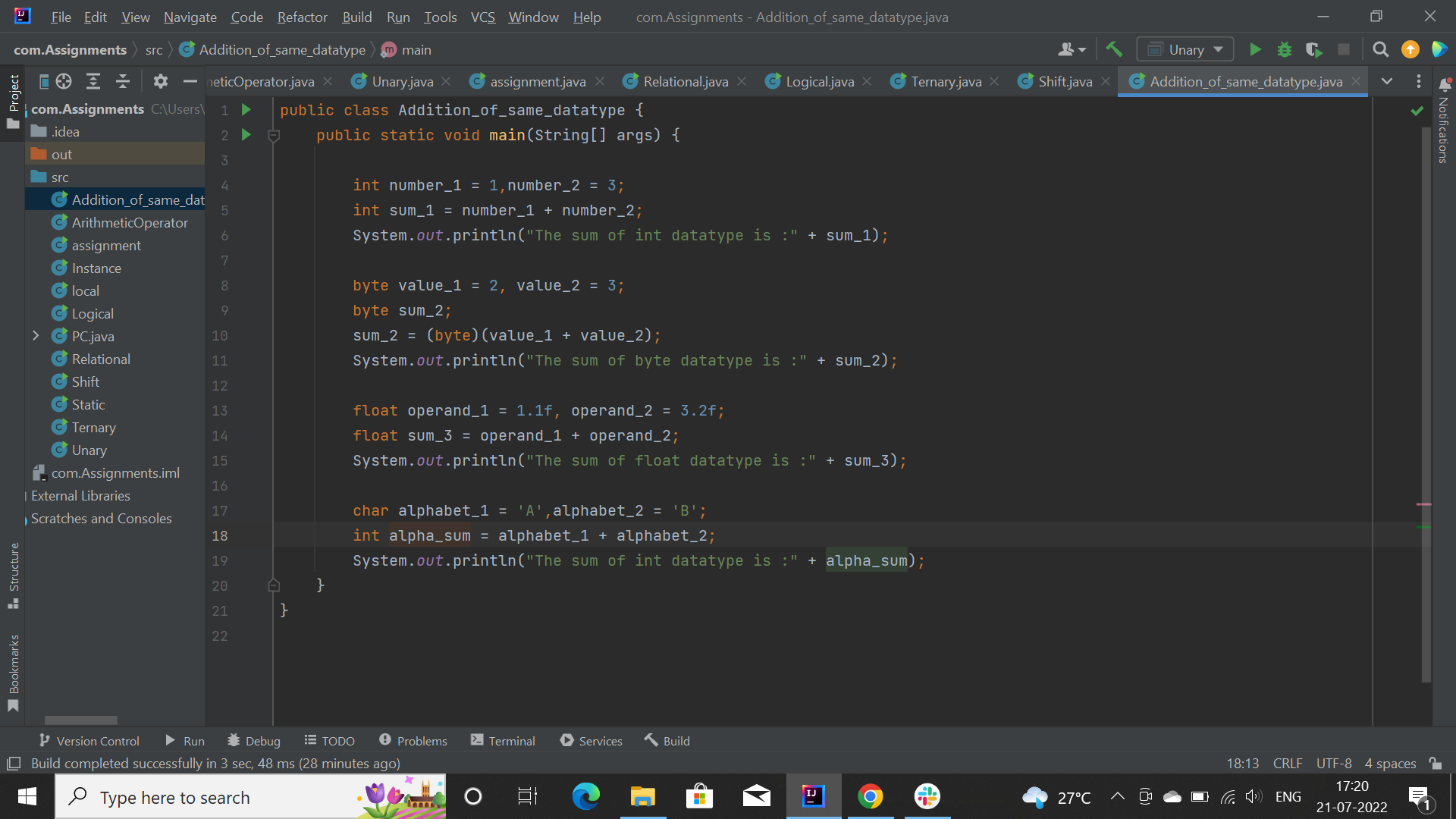
****

**Difference:**

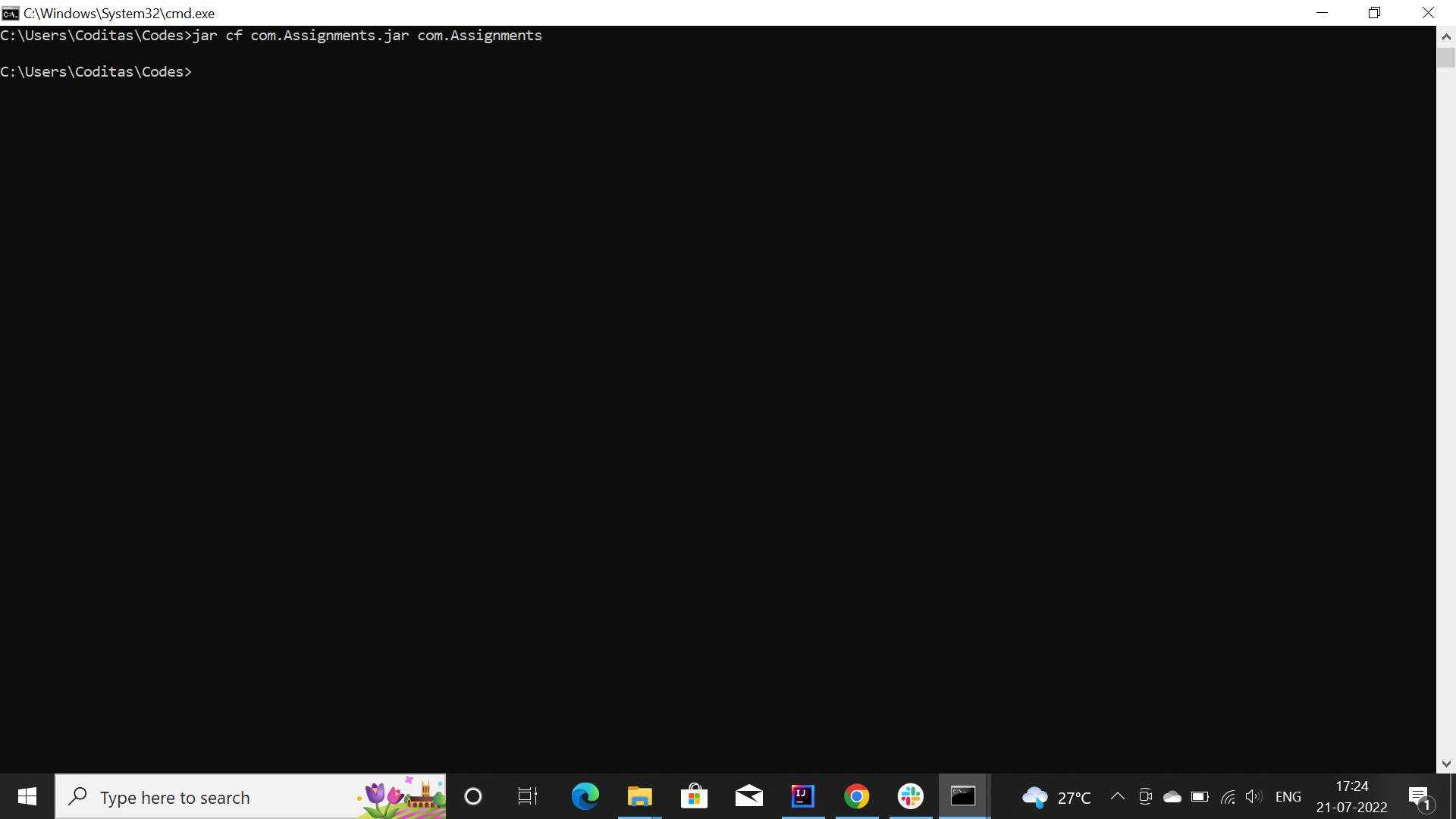
There are a few differences between the bitwise operators we've discussed here and the more commonly known logical operators.

1. First, logical operators work on *boolean* expressions and return *boolean* values (either *true* or *false),* whereas bitwise operators work on binary digits of integer values (*long, int, short, char,* and *byte*) and return an integer.
2. Also, logical operators always evaluate the first *boolean* expression and, depending on its result and the operator used, may or may not evaluate the second. On the other hand, bitwise operators always evaluate both operands.
3. Finally, logical operators are used in making decisions based on multiple conditions, while bitwise operators work on bits and perform bit by bit operations.

**4) Addition of 2 same data type variables:**

****

**5) Create .jar file of your project**

****