**Basic Programming Practicum Experiment Job sheet 5 Meeting 6**



**From:**

AL AZHAR RIZQI RIFA’I FIRDAUS

**Class:**

1 I

**Absence:**

01

**Major:**

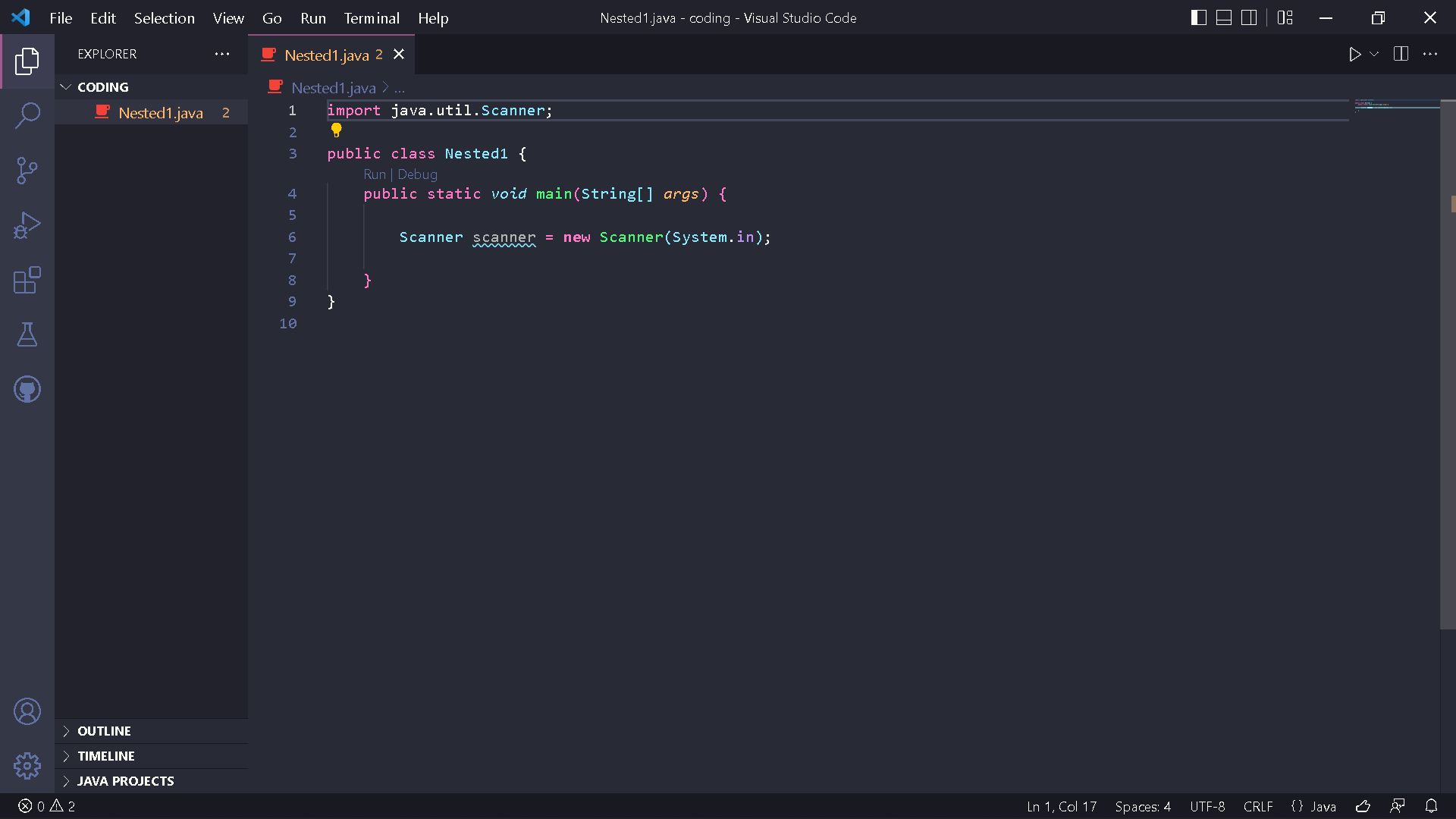
Information Technology

**Study Program:**

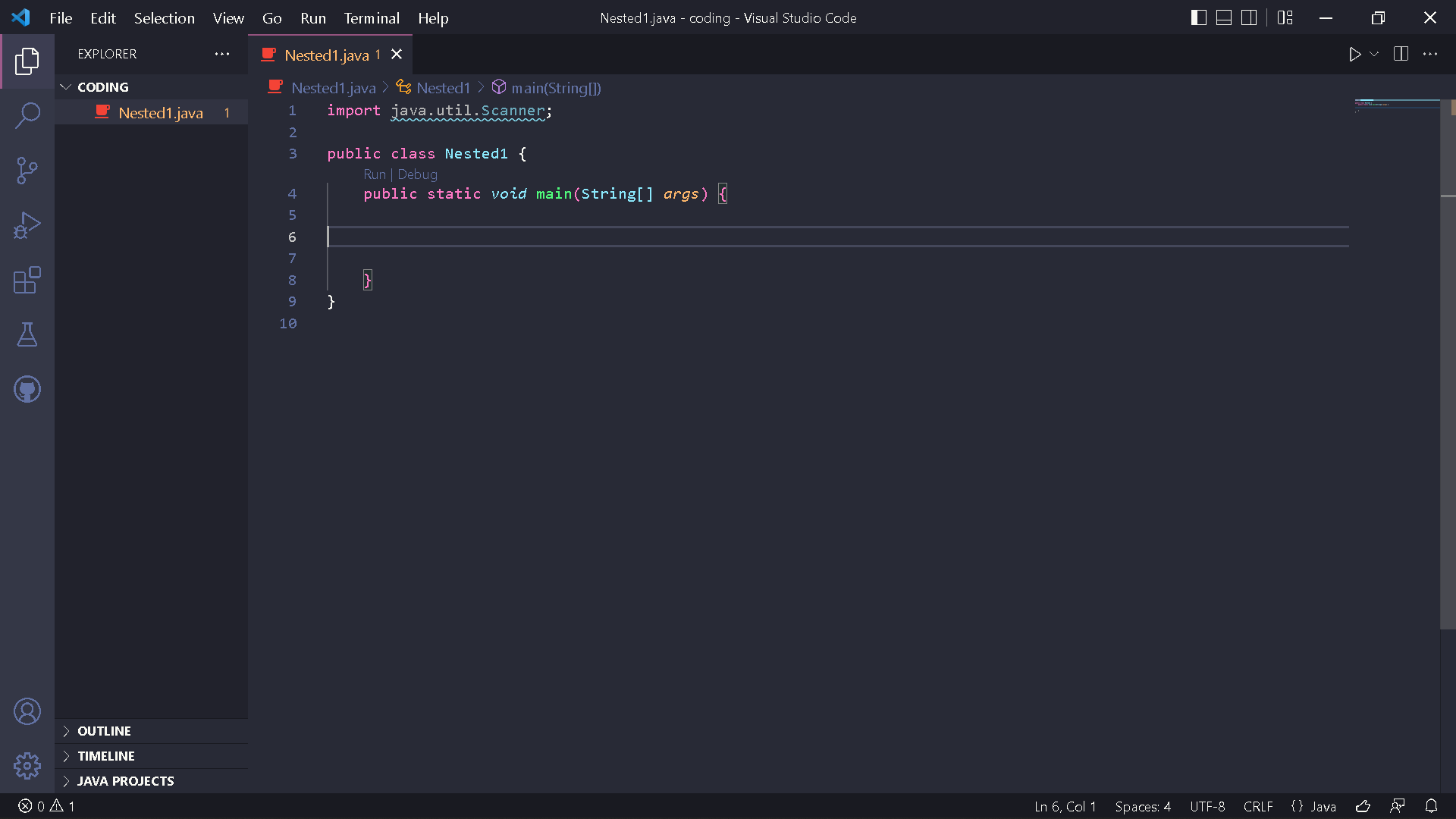
Informatic Engineering

Experiment 1

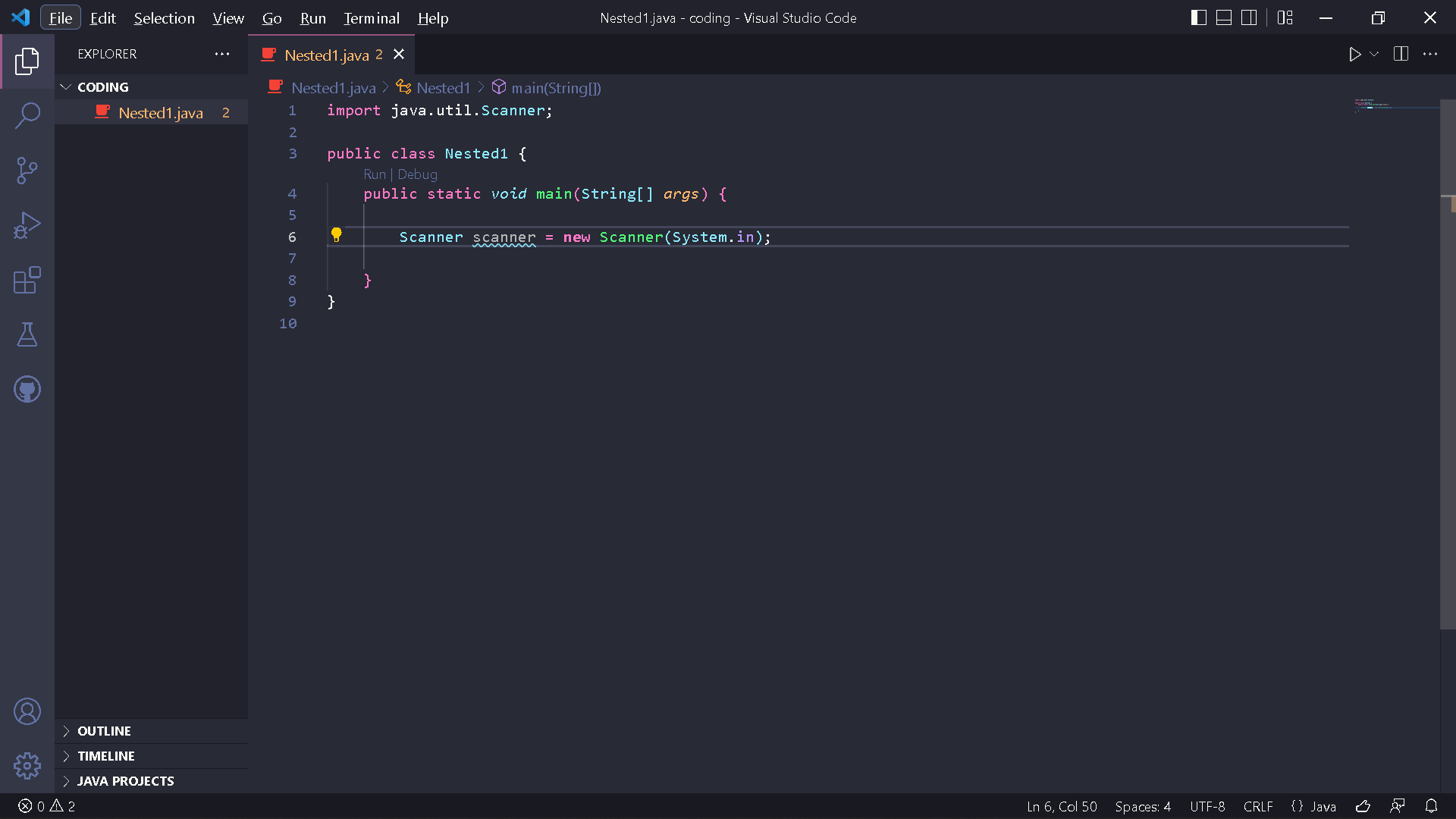
1. Open a text editor. Create a new file, name it Nested1.java



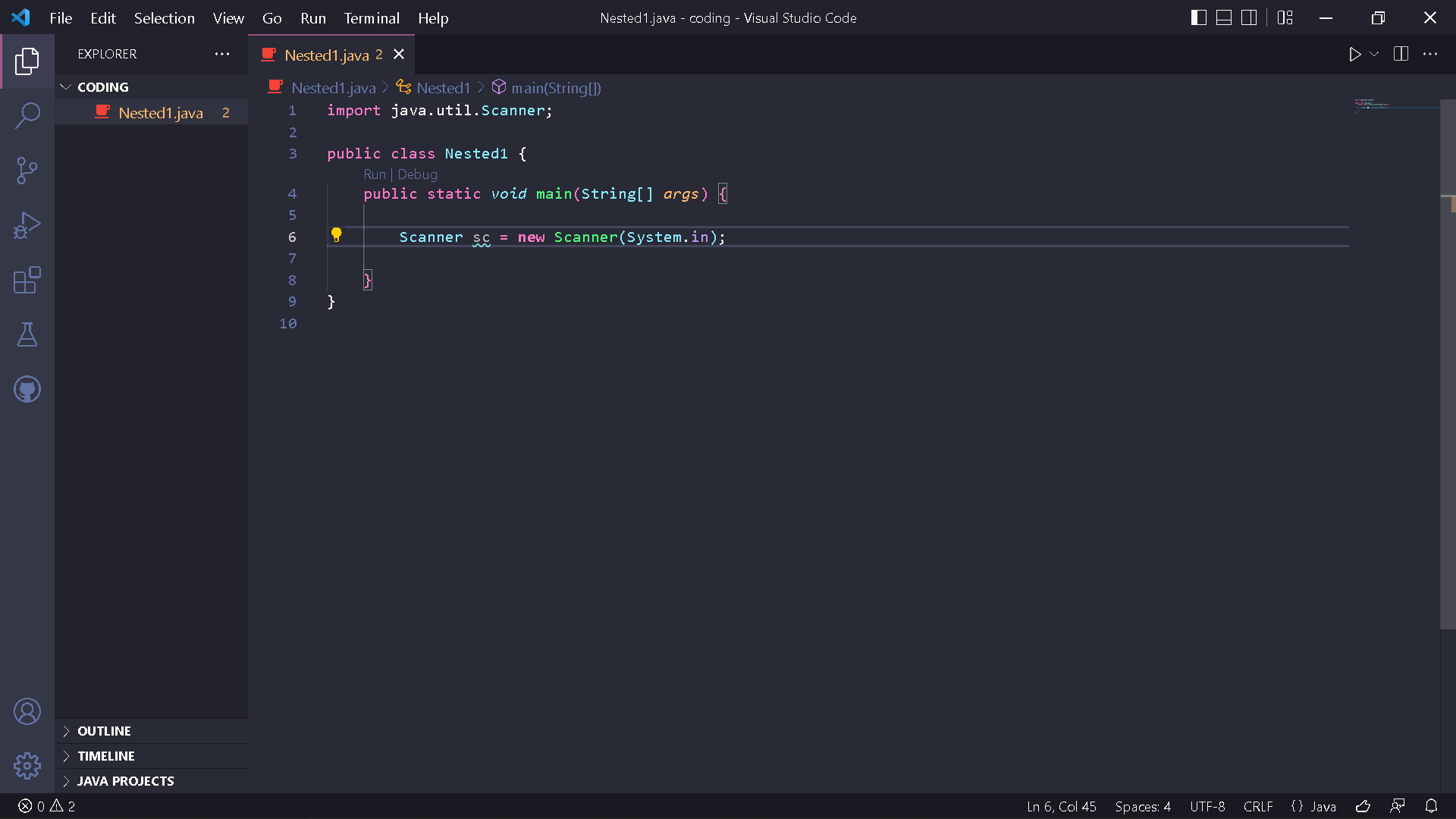
1. Write the basic structure of the Java programming language which contains the main() function



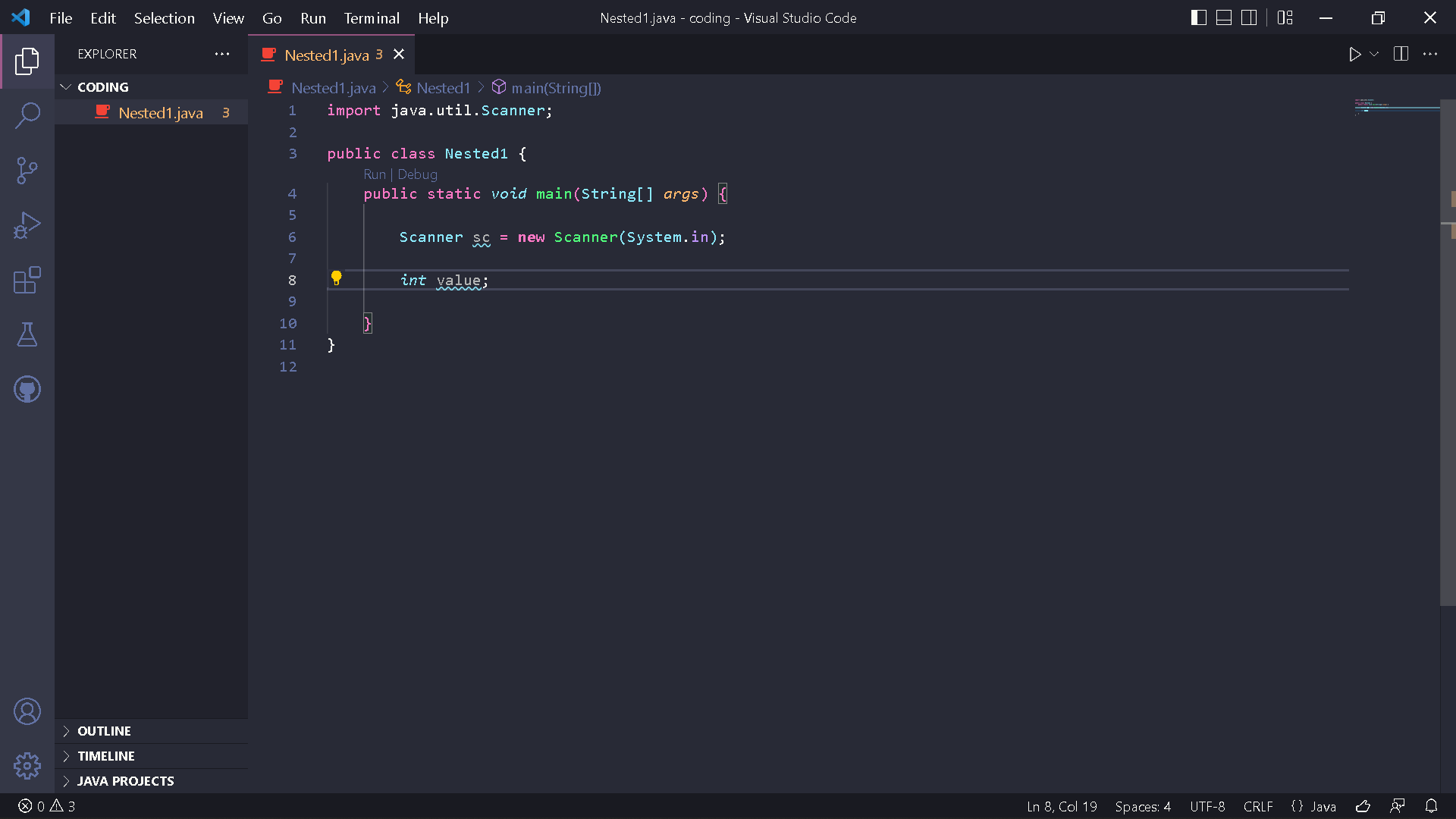
1. Add the Scanner library.



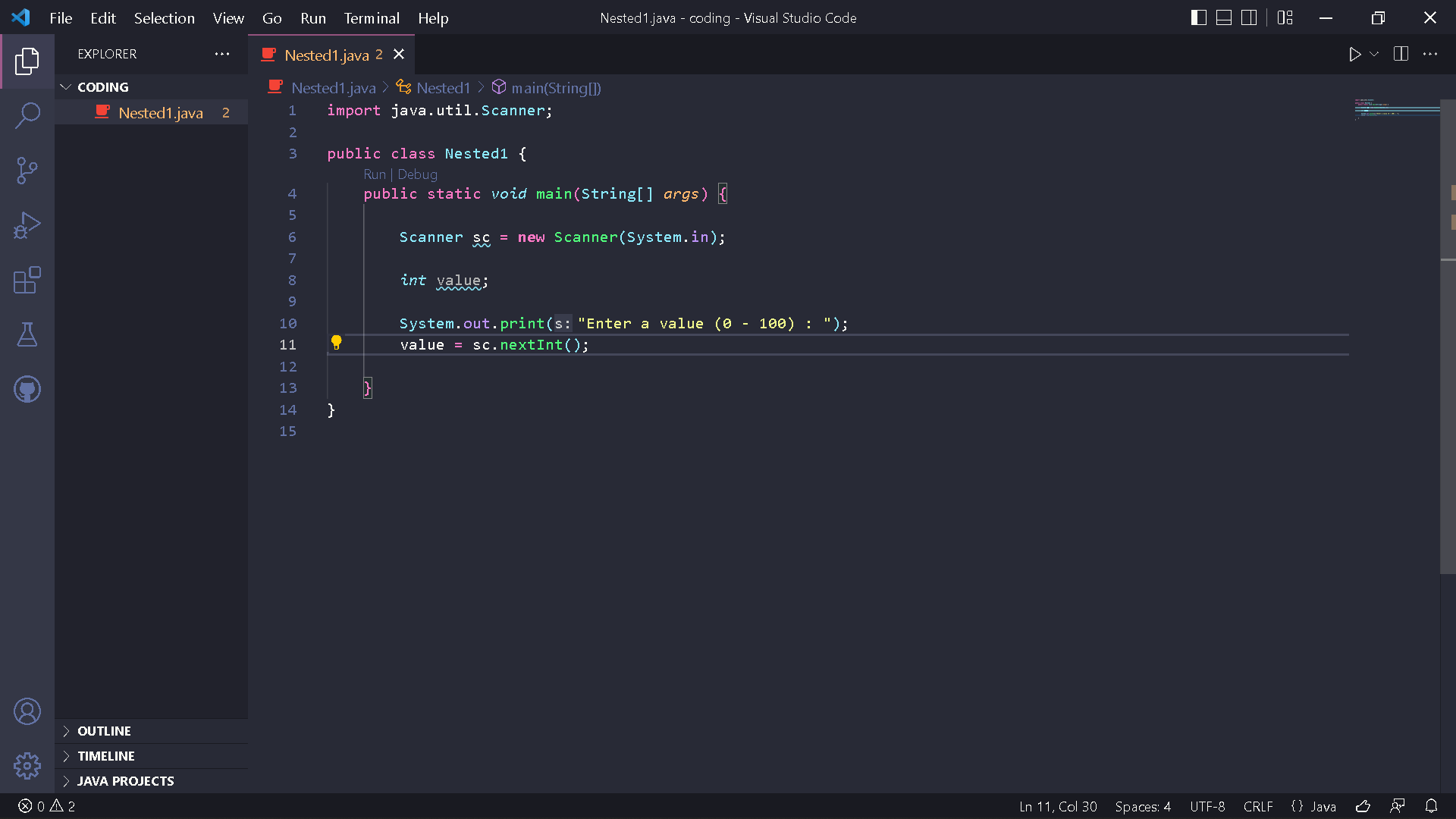
1. Make a Scanner declaration with the name sc



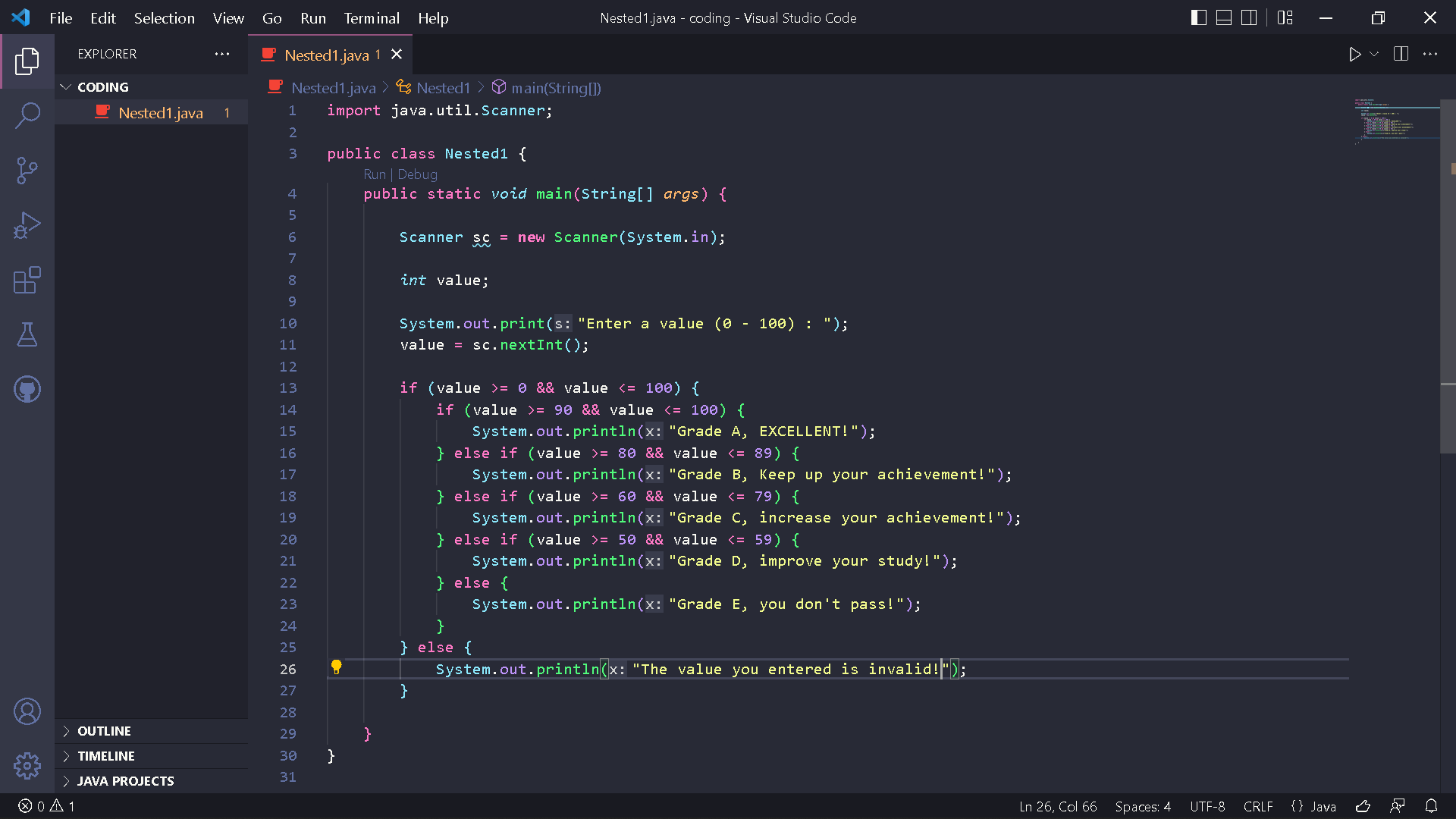
1. Create an int variable with the name value



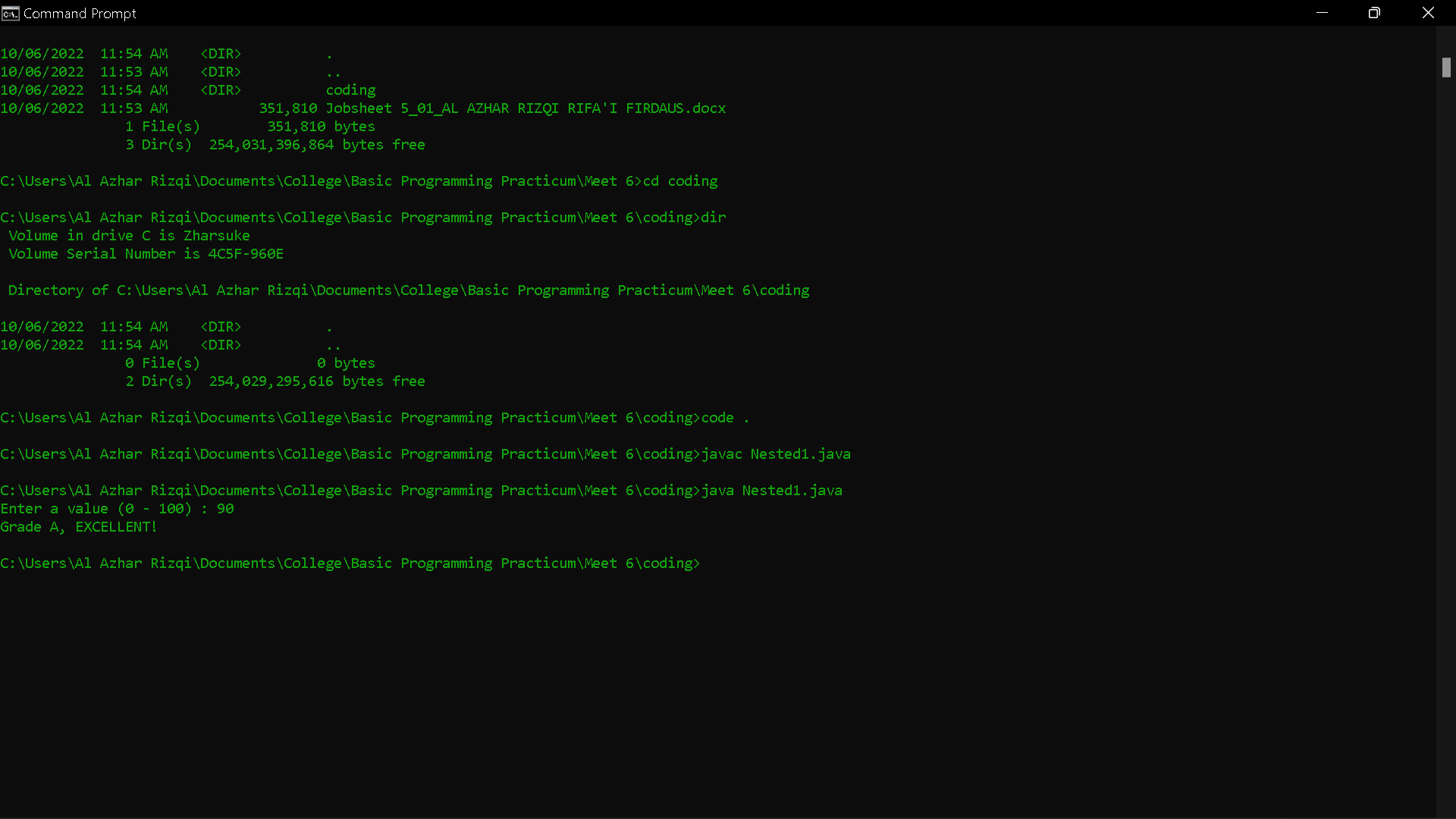
1. Write down the syntax for entering the value from keyboard



1. Create a nested selection structure. The first check is used to ensure that the value entered is in the range 0 - 100. If the value is in the range 0 - 100, then a student graduation status will be checked, i.e. if the value is between 90 - 100 then the value is A, if the value is between 80 - 89 then the value is B, if the value is between 60 - 79 then the value is C, if the value is between 50 - 59 then the value is D, and if the value is between 0 - 49 then the value is E. Whereas if the value is outside the range 0 - 100 , then displayed information stating that the value entered is invalid.



1. Compile and run the program. Observe the results!



Questions!

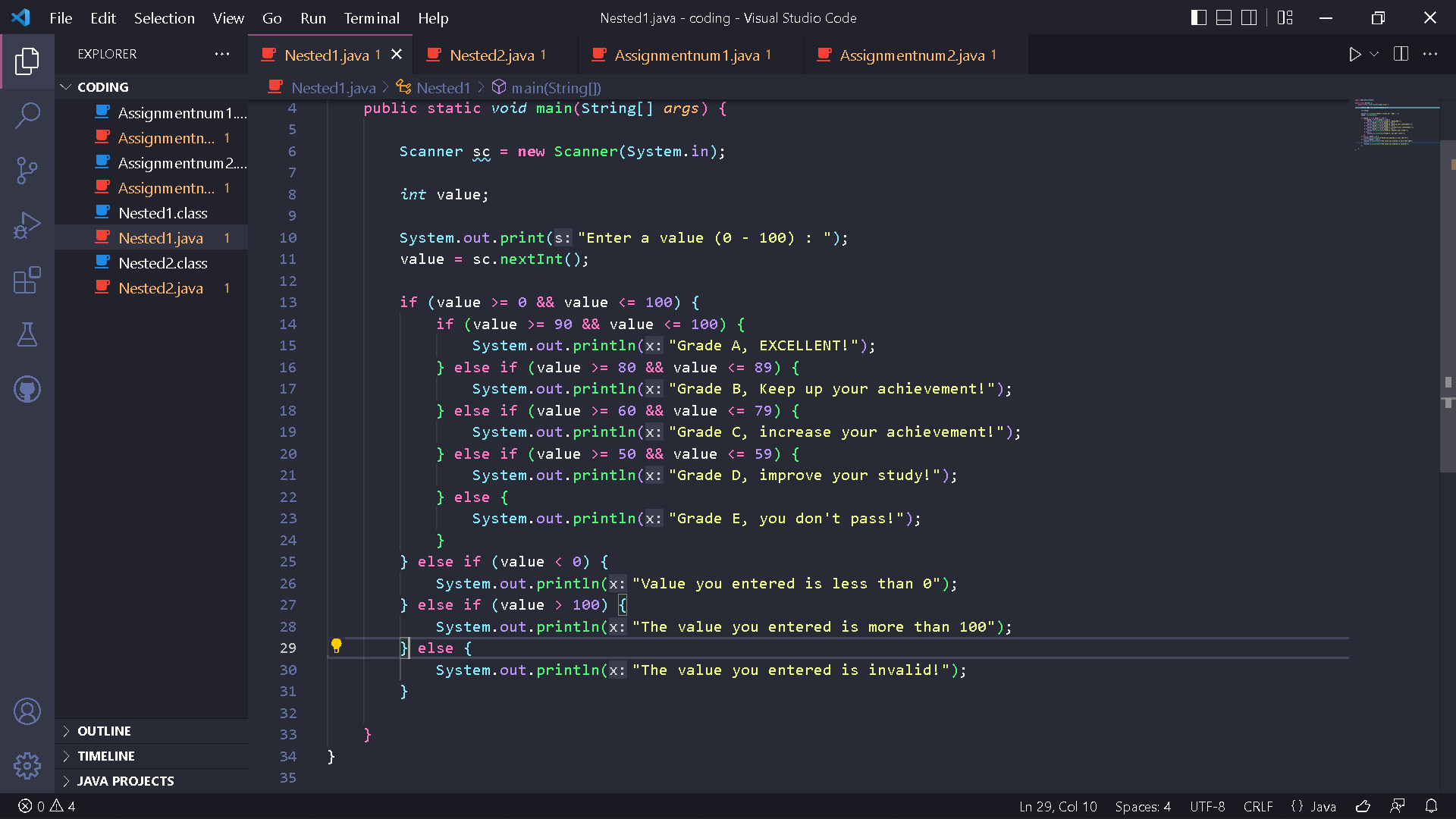
1. Describe the following syntax functions! if (value >= 0 && value <= 100)

2. Modify the program code in Experiment 1 so that if the entered value is less than 0 the output "Value you entered is less than 0" and if the entered value is more than 100 the output will display "The value you entered is more than 100"!

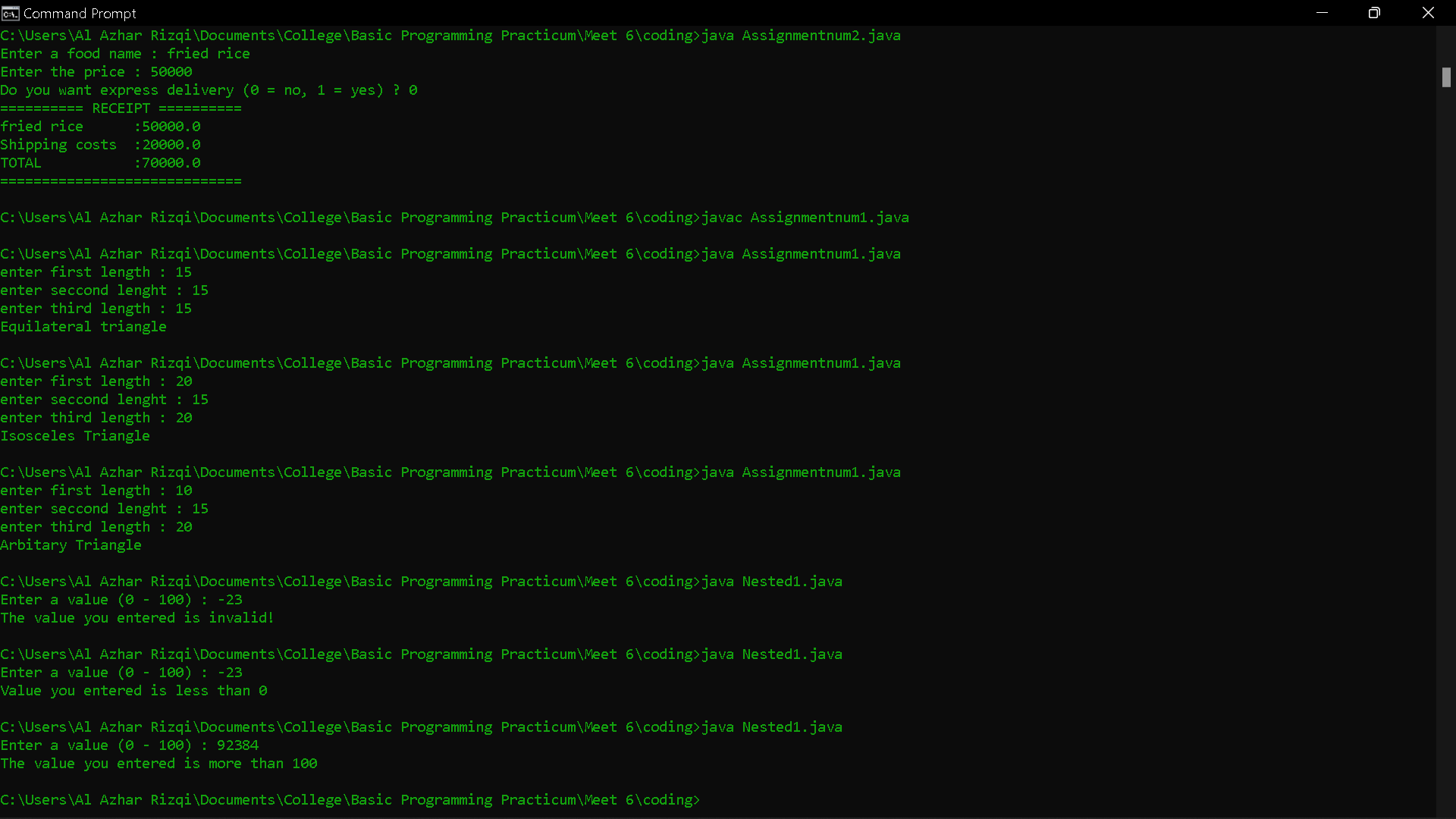
3. Change the && operator to || on if (value >= 0 && value <= 100). Compile and run the program by entering the value = 105 using keyboard. Watch what happened! Why is the result like that?

Answer!

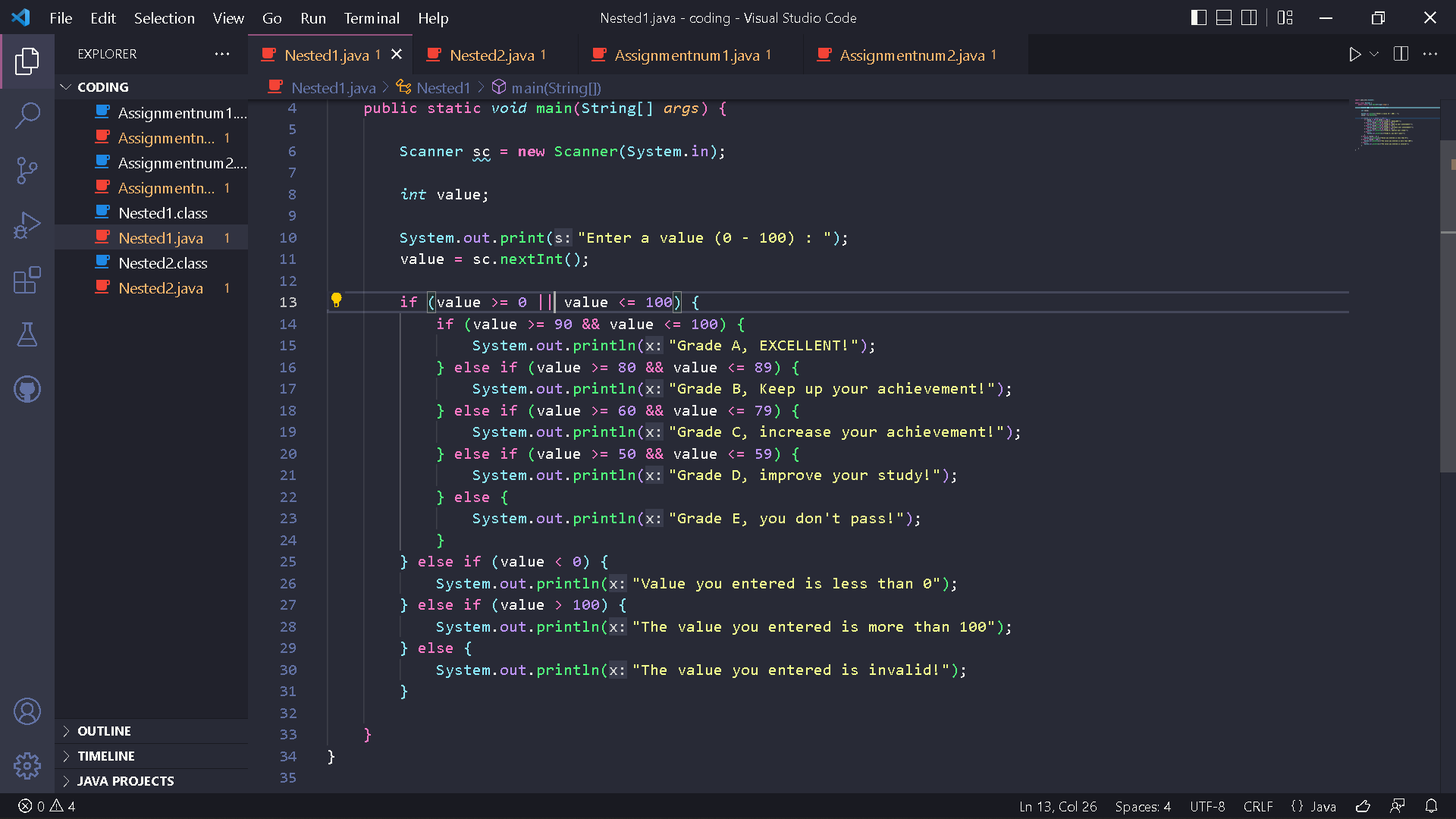
1. Its function is to combine the shading between more than equal to 0 and less than equal to 100 which results in 0 to 100
2. Code :



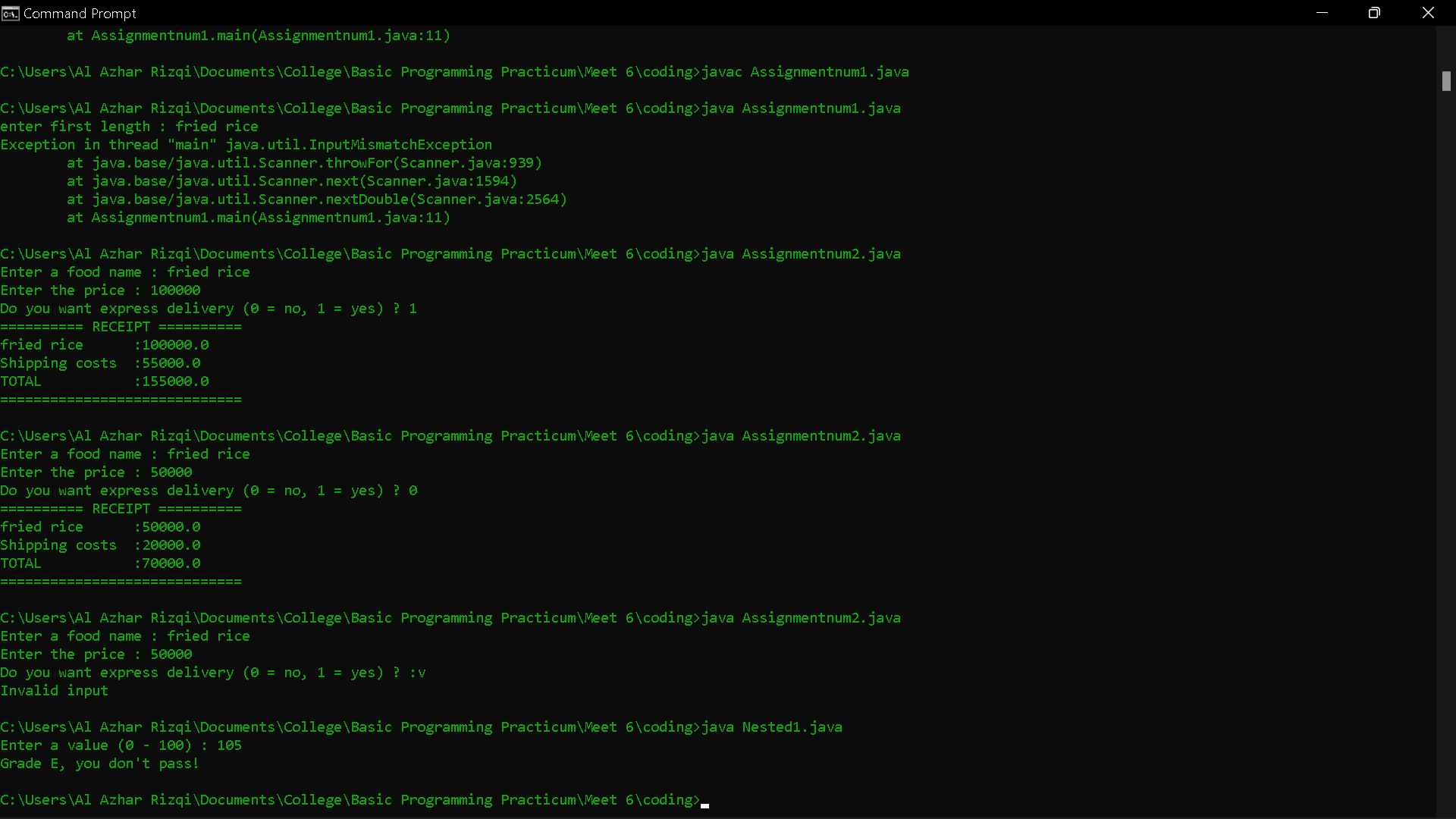
Result :



1. Code :



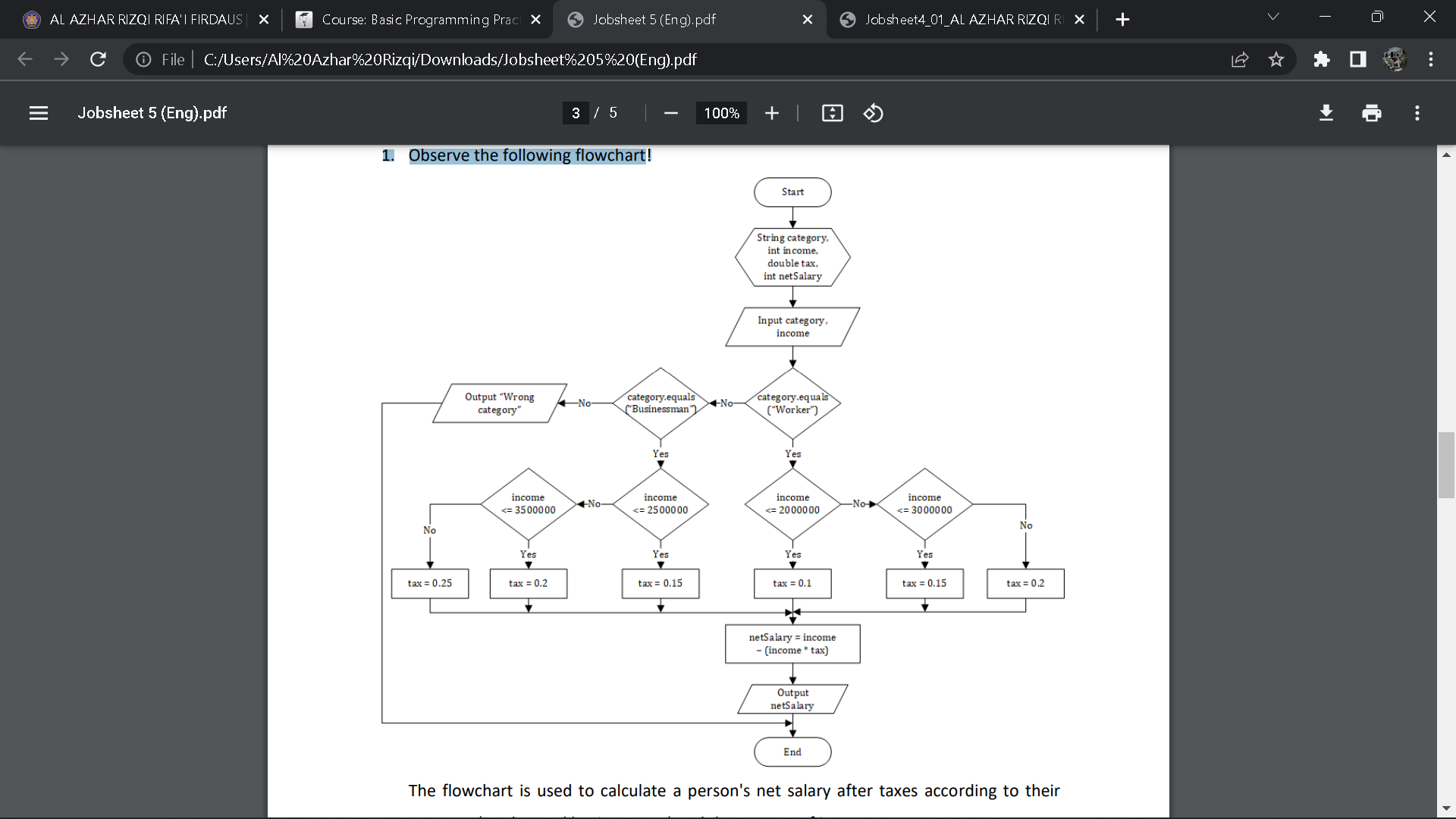
Result :



Because if or only forms 2 shading numbers, it doesn't combine shading numbers.

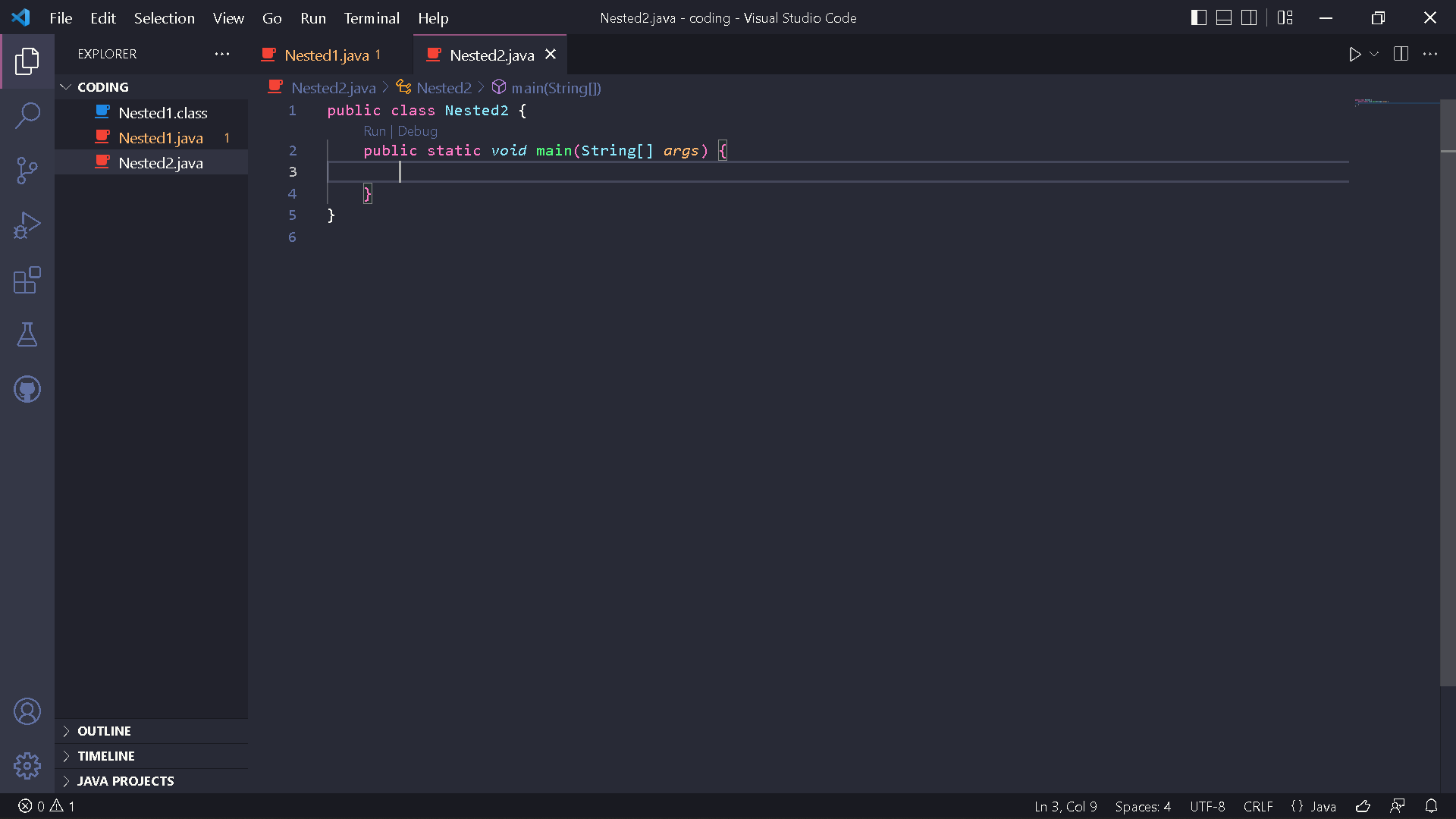
Experiment 2

1. Observe the following flowchart!

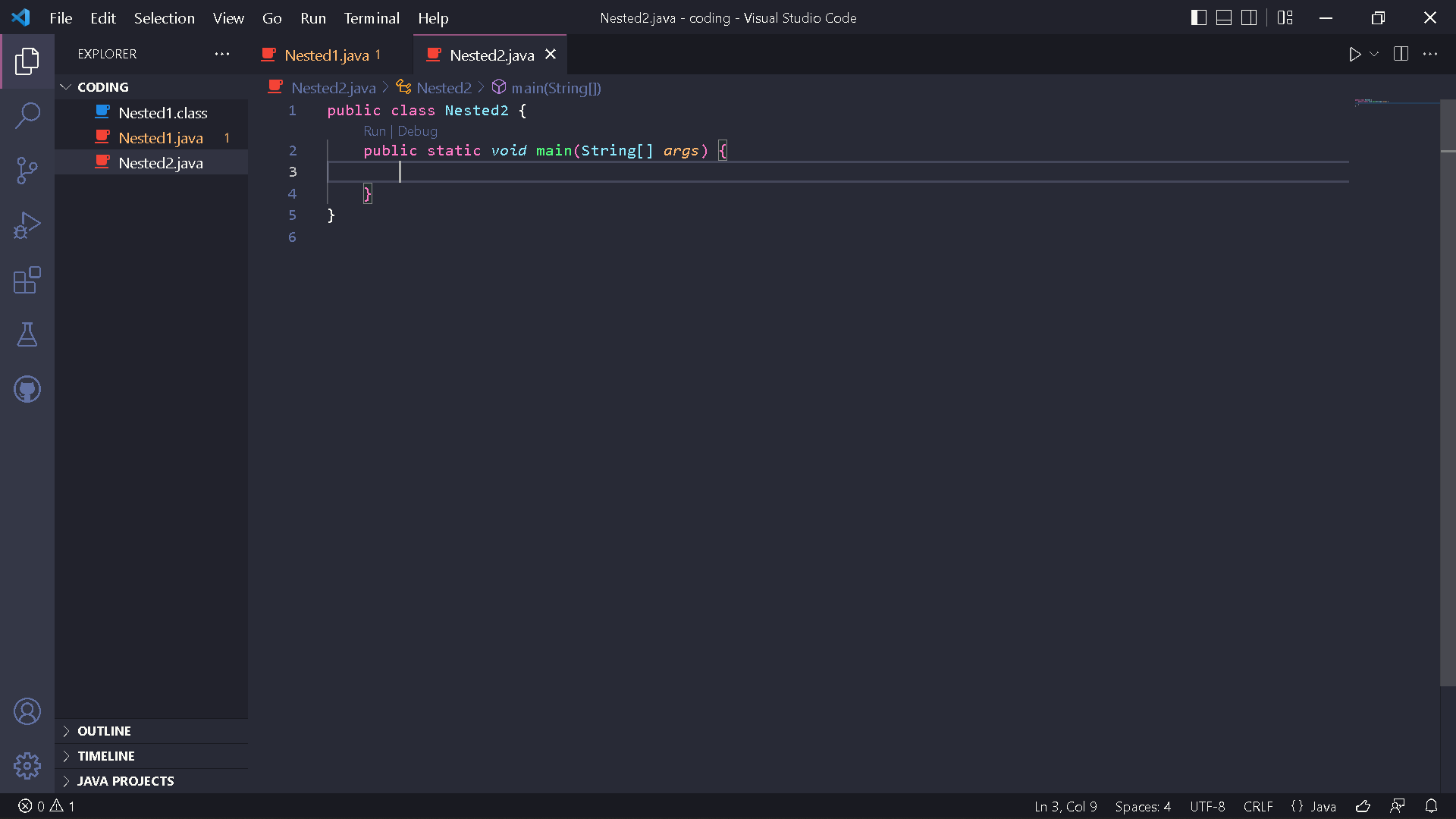


The flowchart is used to calculate a person's net salary after taxes according to their category (worker and businessman) and the amount of income.

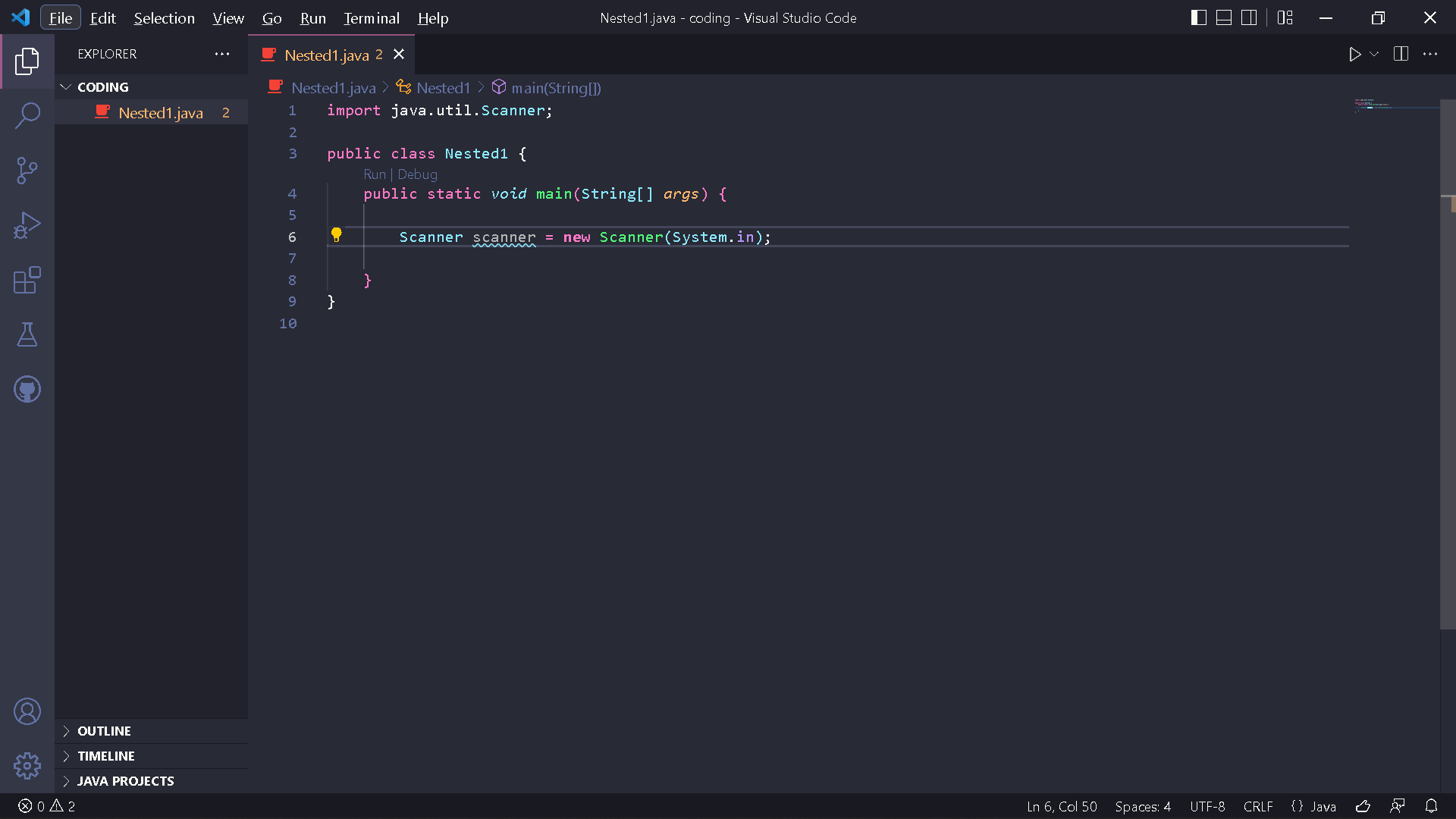
1. Open a text editor. Create a new file, name it Nested2.java



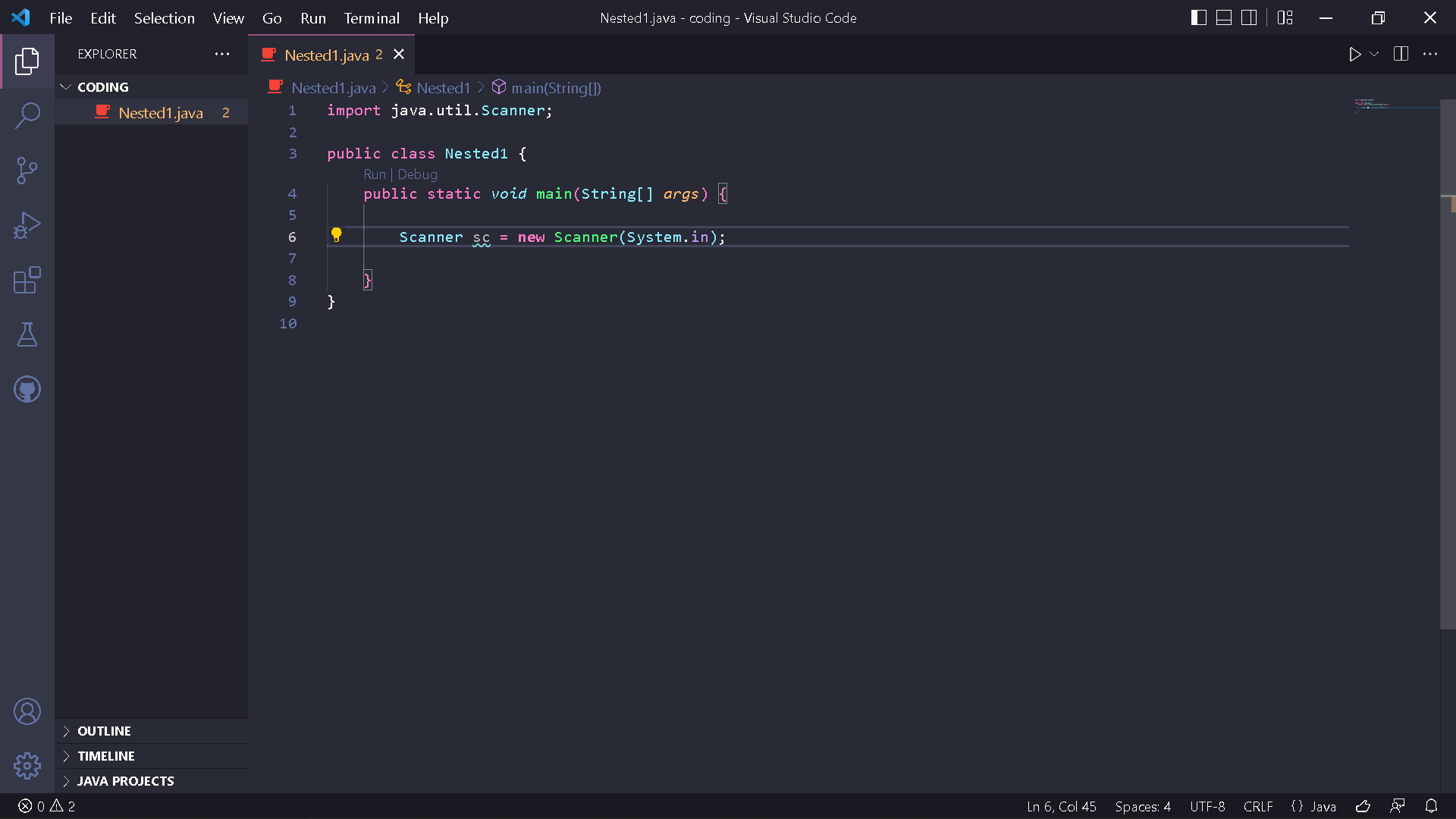
1. Write the basic structure of the Java programming language which contains the main() function



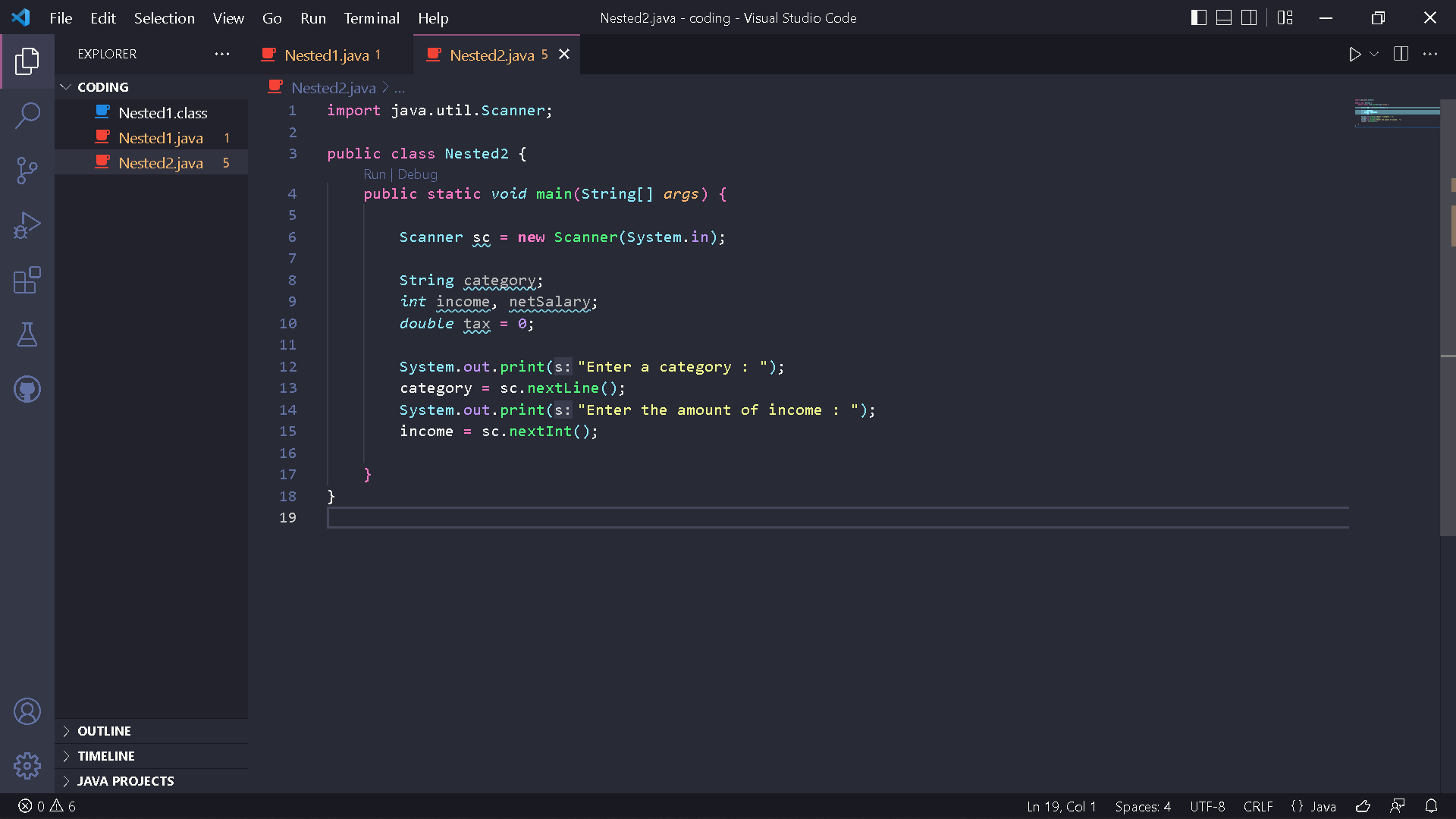
1. Add the Scanner library.



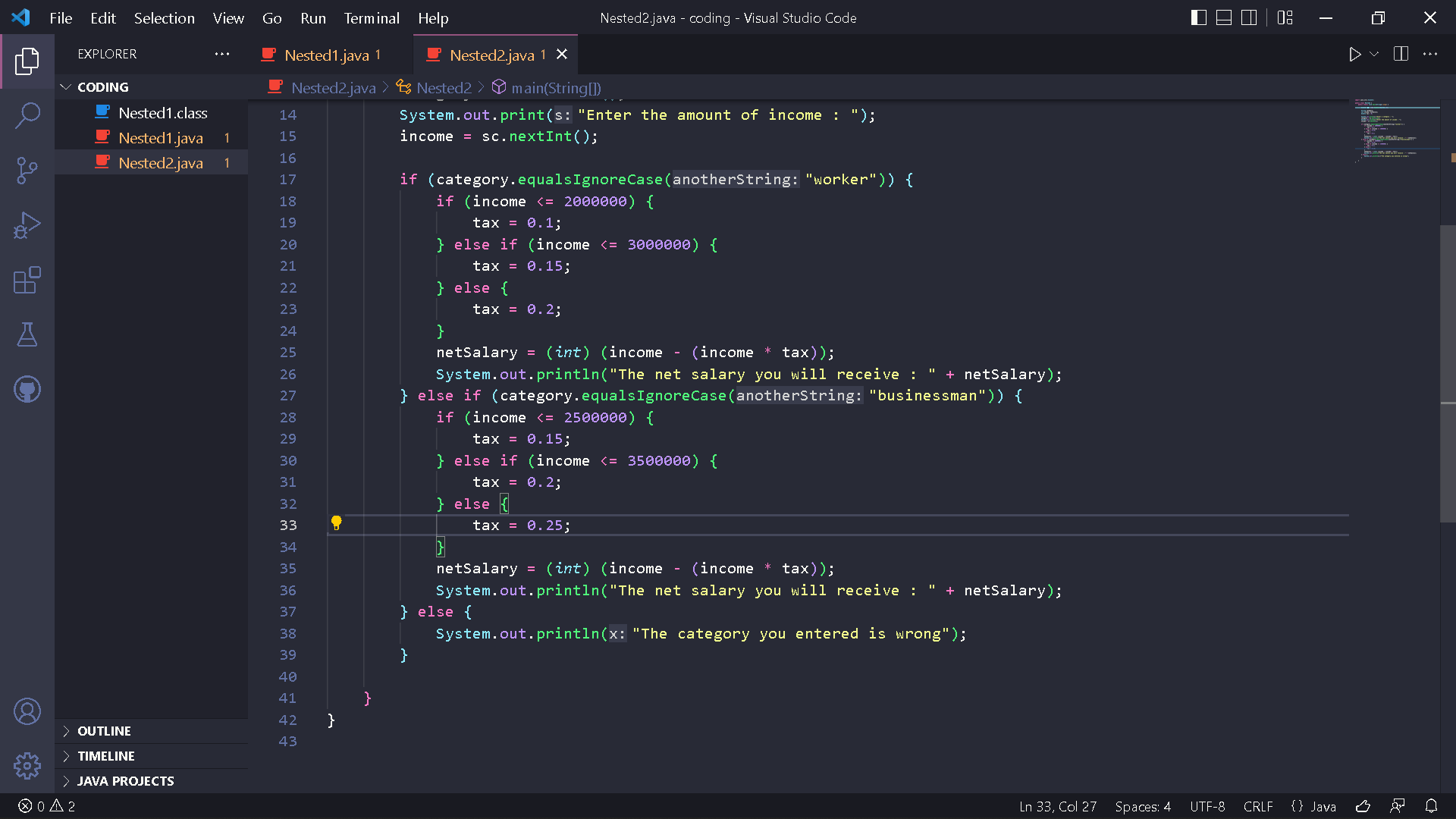
1. Make a Scanner declaration with the name sc



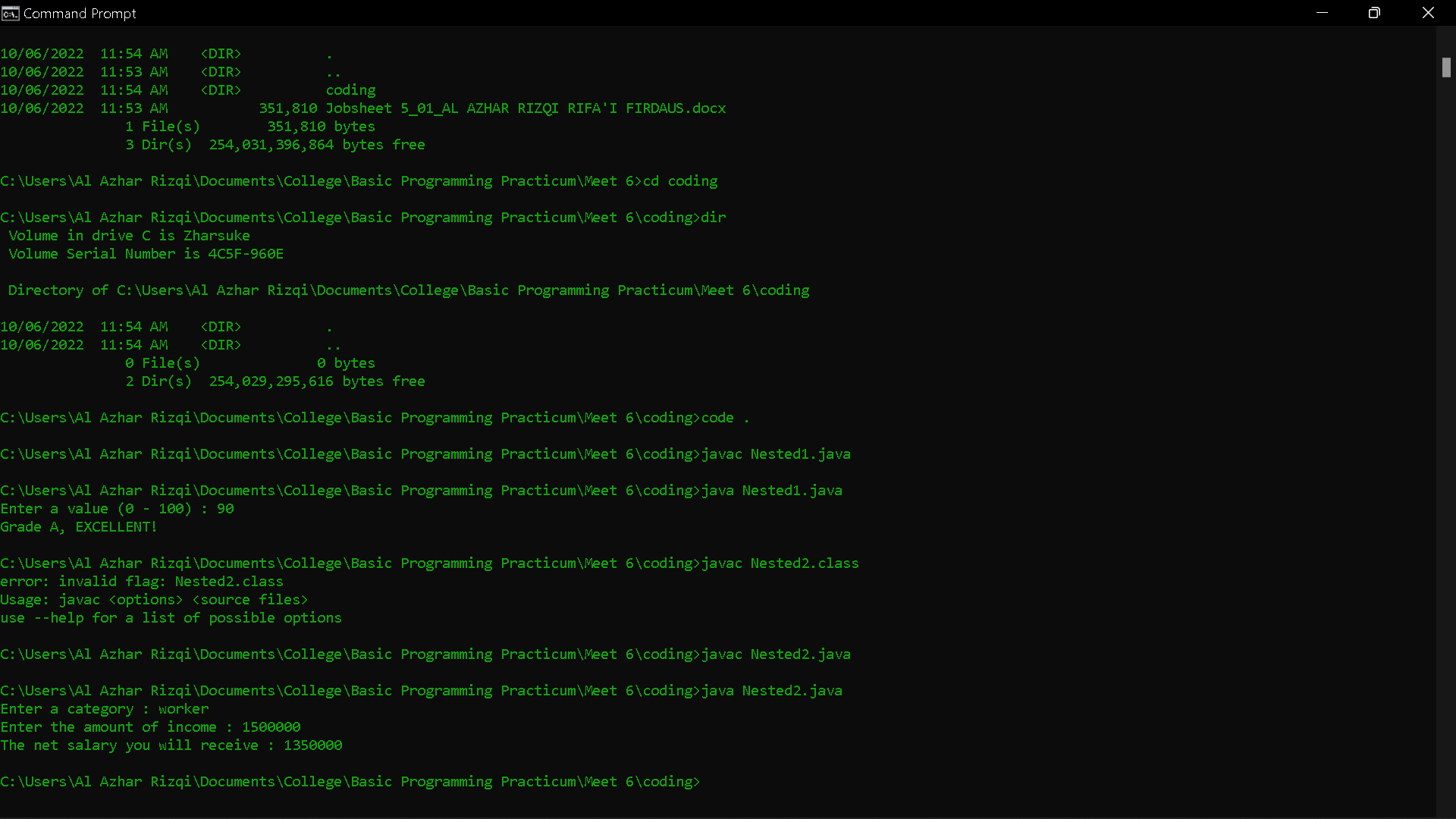
1. Declare category, income, netSalary, and tax variables



1. Write down the syntax for entering the value from keyboard
2. Create a nested selection structure. The first check is used to check the category (worker or businessman). Then a second check is carried out to determine the amount of tax based on the income that has been entered. Then add the program code to calculate the net salary received after taxes



1. Compile and run the program. Observe the results!



Questions!

1. Run the program by entering category = worker and income = 2048485 using keyboard. Watch what happened! Why is the decimal number not displayed?

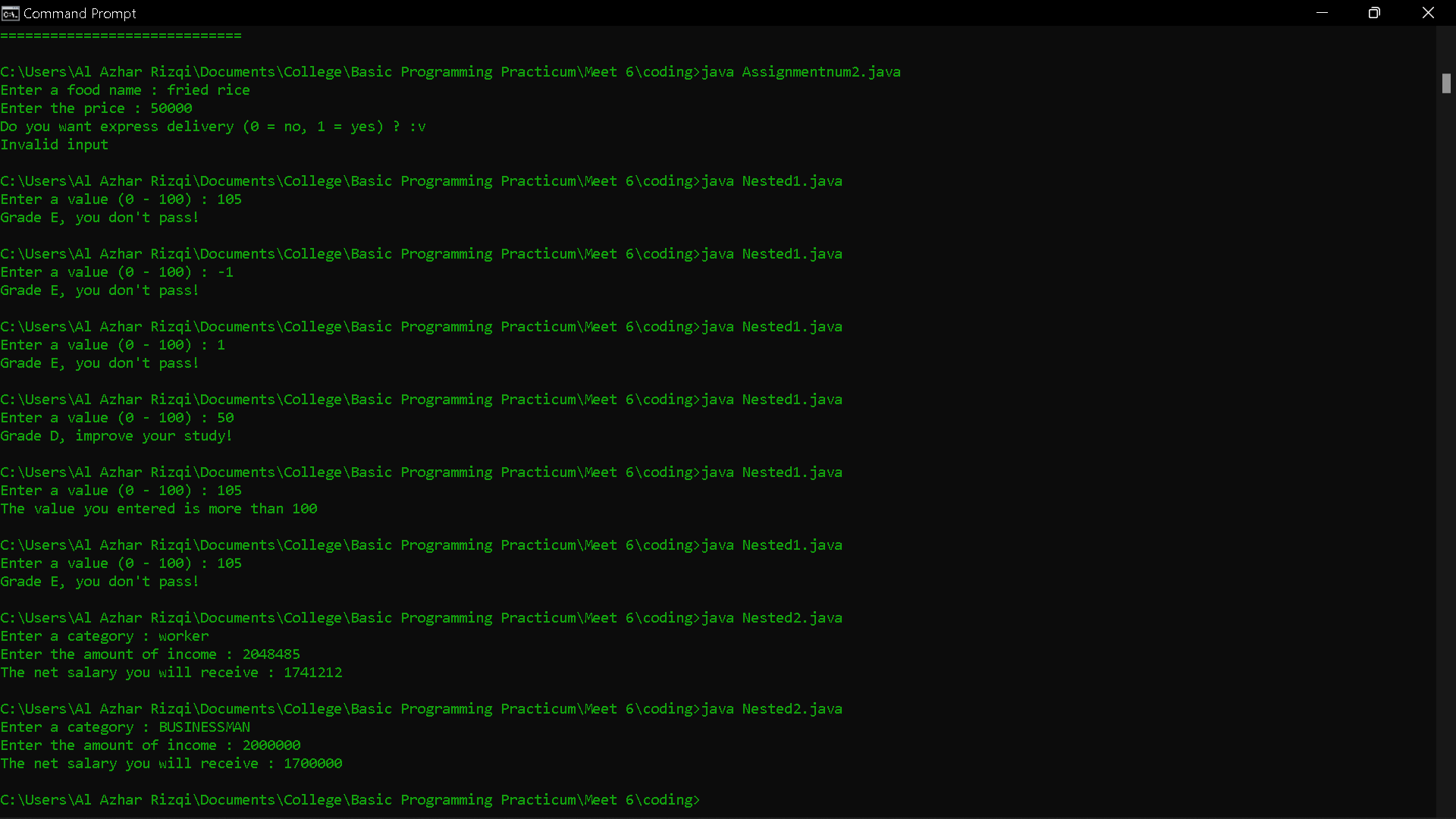
2. Describe the function of (int) in the following syntax! netSalary = (int) (income - (income \* tax));

3. Run the program by entering category = BUSINESSMAN and income = 2000000. Watch what happens! What are the uses of equalsIgnoreCase?

4. Change equalsIgnoreCase to equals, then run the program by entering category = BUSINESSMAN and income = 2000000. Watch what happens! Why is the result like that? What are the uses of equals?

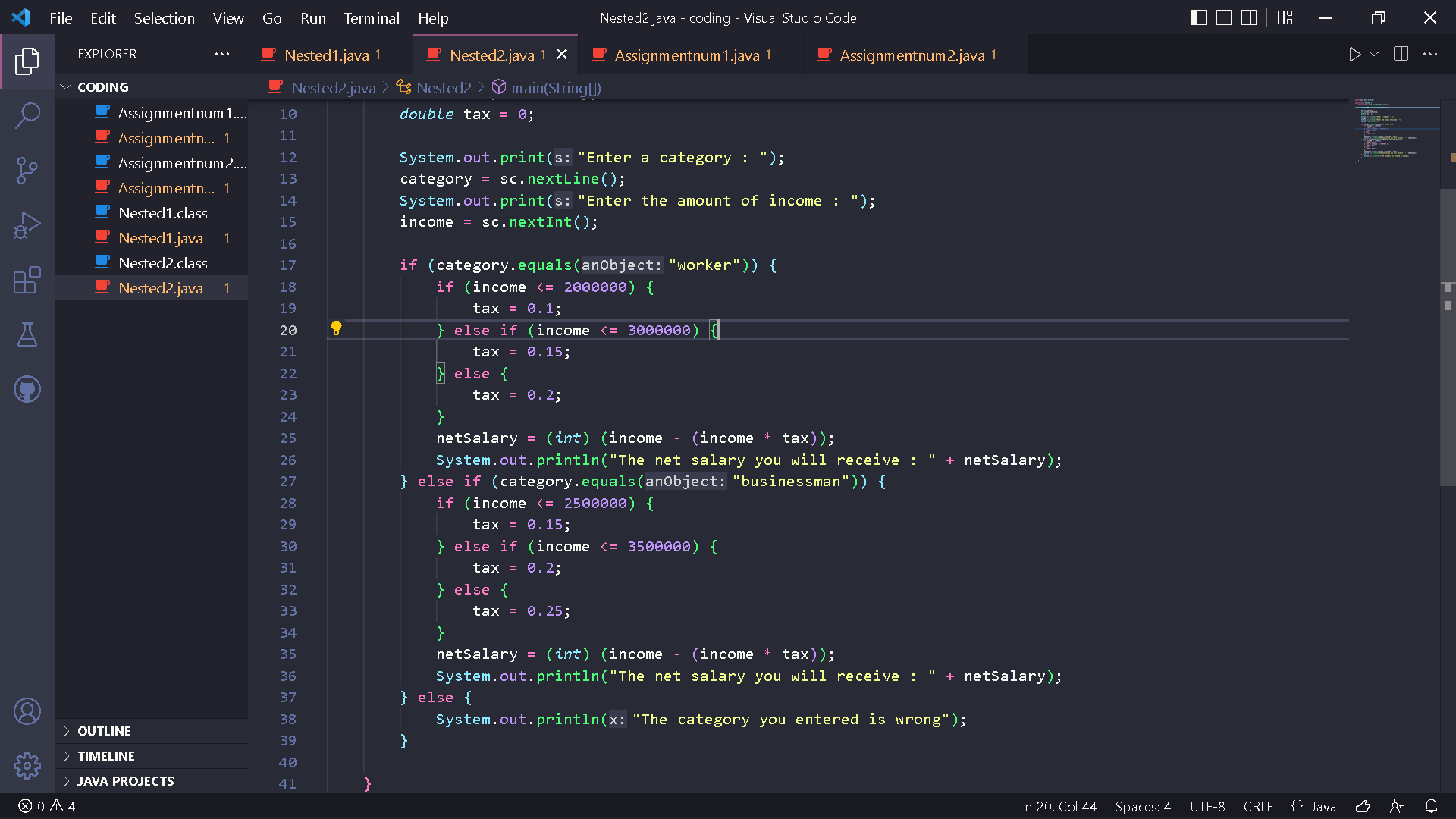
Answer!

1. Because the data type of the netSalary variable is integer not double or float whose function is to display decimal numbers.
2. The function from (int) to casting variable besides int to int.
3. Result :

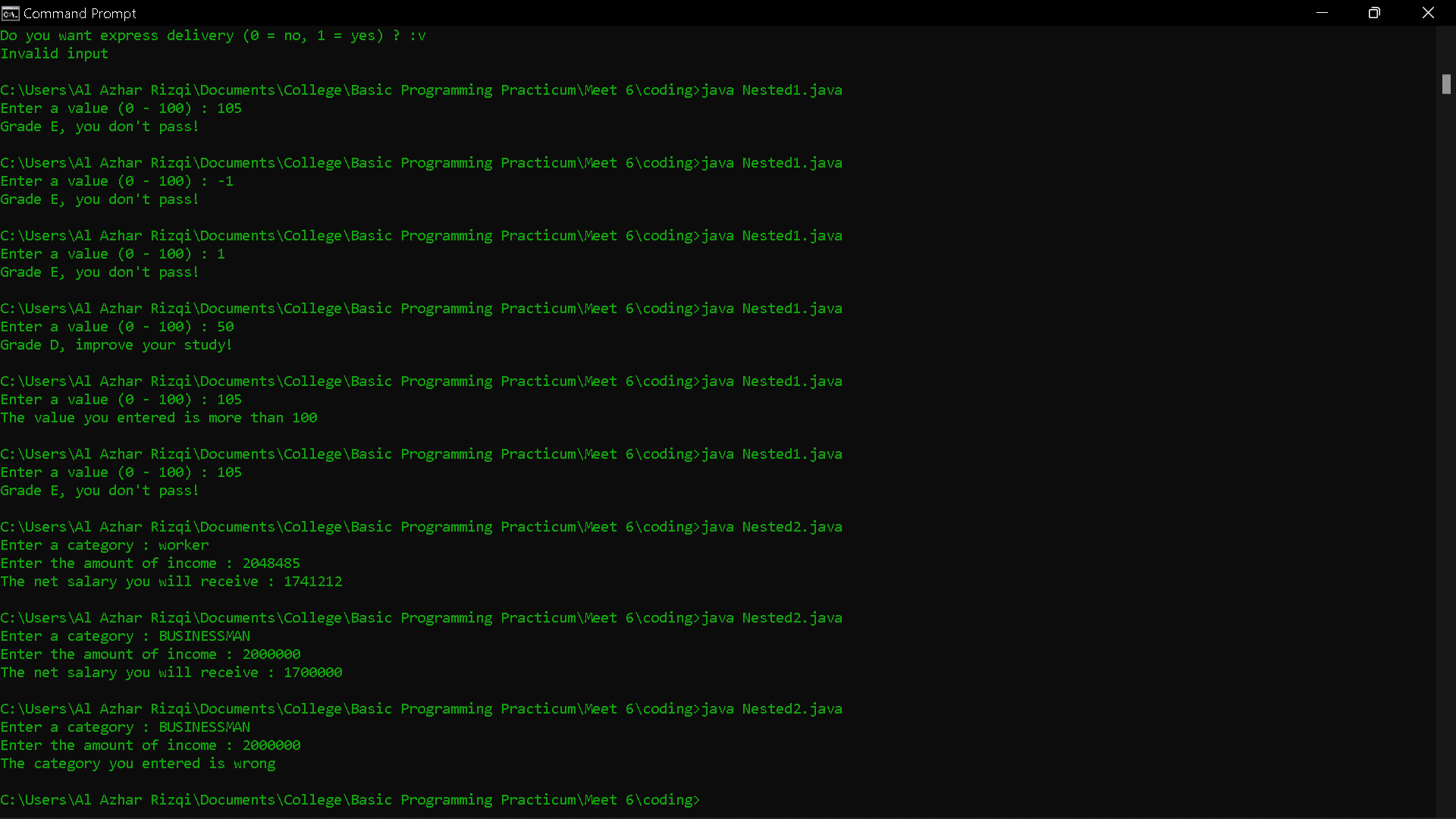


Uses of equalsIgnoreCase is method compares two strings, ignoring lower case and upper case differences.

1. Code :



Result :



Because without ignorecase, the method doesn't ignore the difference between lowercase and uppercase letters. The uses of equals only compare if it is equal to, nothing more.