class ----->  class ======> extends  ====> no multiple inheritance  
interface ----> class =====> Implements  
interface ---> interface ====> extends

Exceptions:   --> unexpected situations

  2 categoried  
    - Checked  ---> compile time  
    - Unchecked  ---> Runtime time..

  pre defined exception classes  
    FileNotFound  
    ArrayIndexoutofBounds exception  
    NullPointer Exception  
    Arthematic Exception ...  
  user defined Exceptions  
try catch blocks  
  try{

  }catch(Exception ex){

  }catch(ArthematicException ex1{  
  }  
  finally{

   sout(" ");  
  }

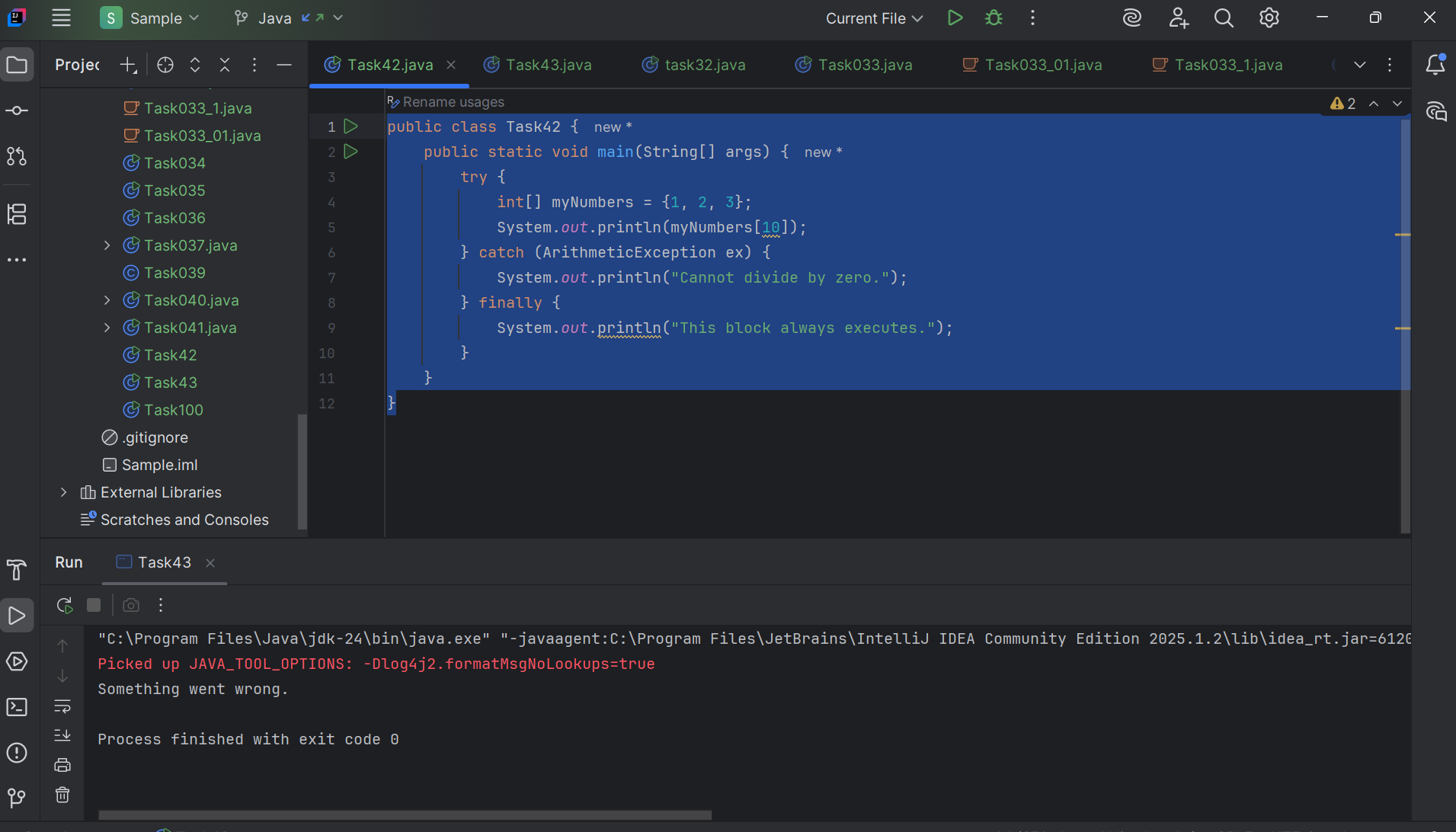
 Day 9 - 13th June 2025

|  |
| --- |
| Inheritance, Polymorphism, Encapsulation, Abstraction, Interfaces, Exception Handling, |
| Collections Framework intro, Streams, File I/O, Multithreading overview |

Exceptions:

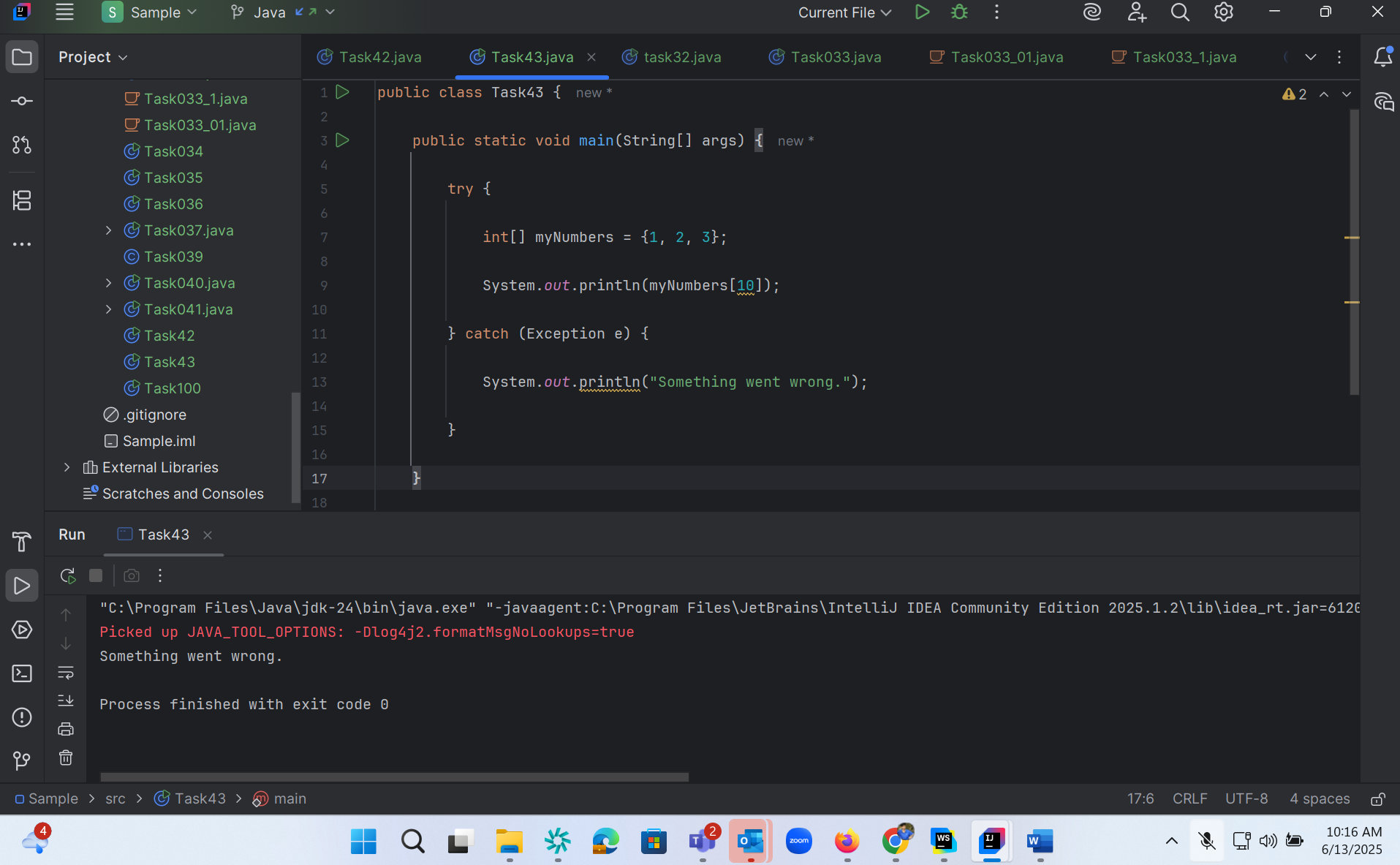
Task 1:

public class Main {  
  public static void main(String[] args) {  
    int[] myNumbers = {1, 2, 3};  
    System.out.println(myNumbers[10]);  
  }  
}



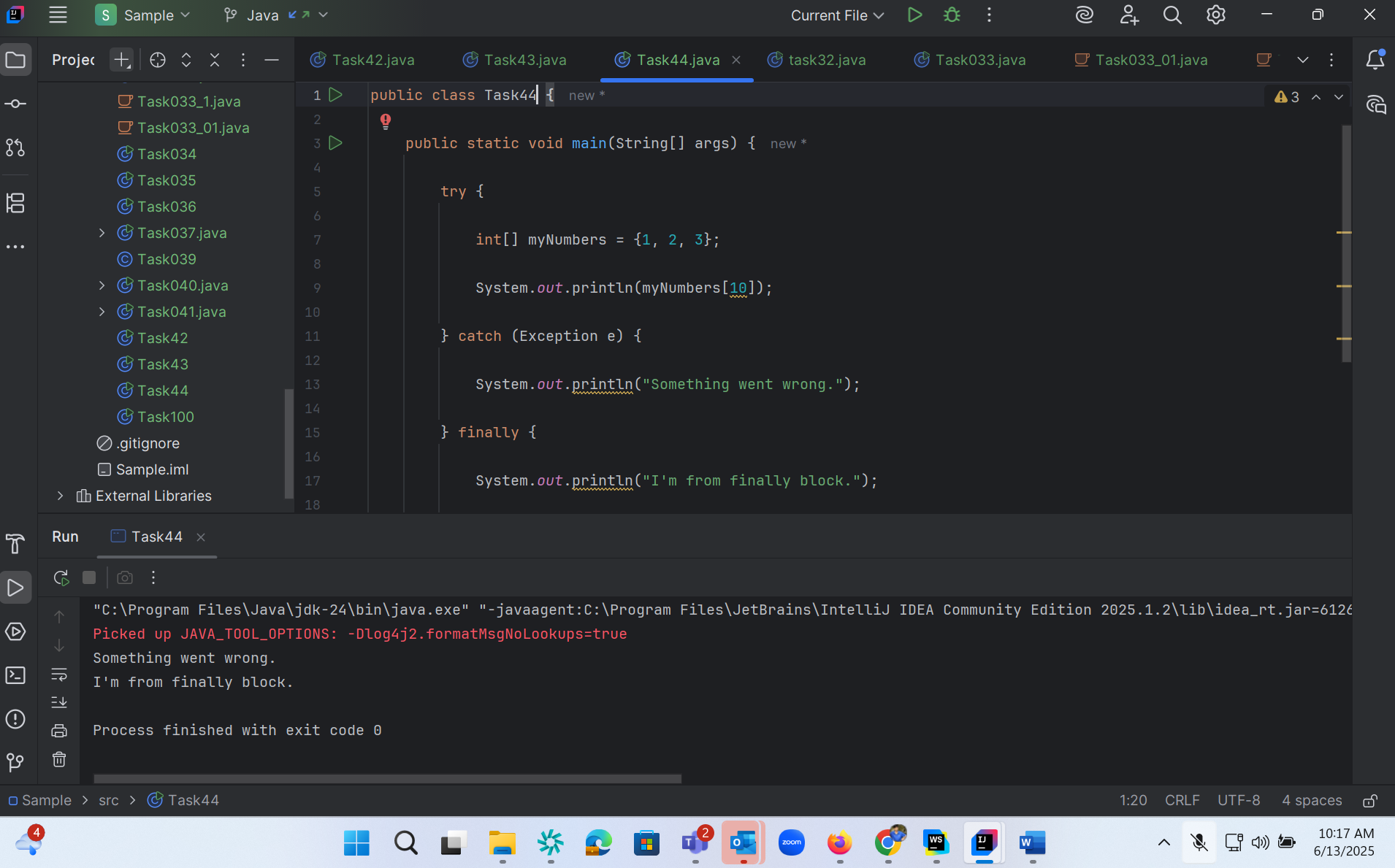
Task 2

public class Main {  
  public static void main(String[] args) {  
    try {  
      int[] myNumbers = {1, 2, 3};  
      System.out.println(myNumbers[10]);  
    } catch (Exception e) {  
      System.out.println("Something went wrong.");  
    }  
  }  
}

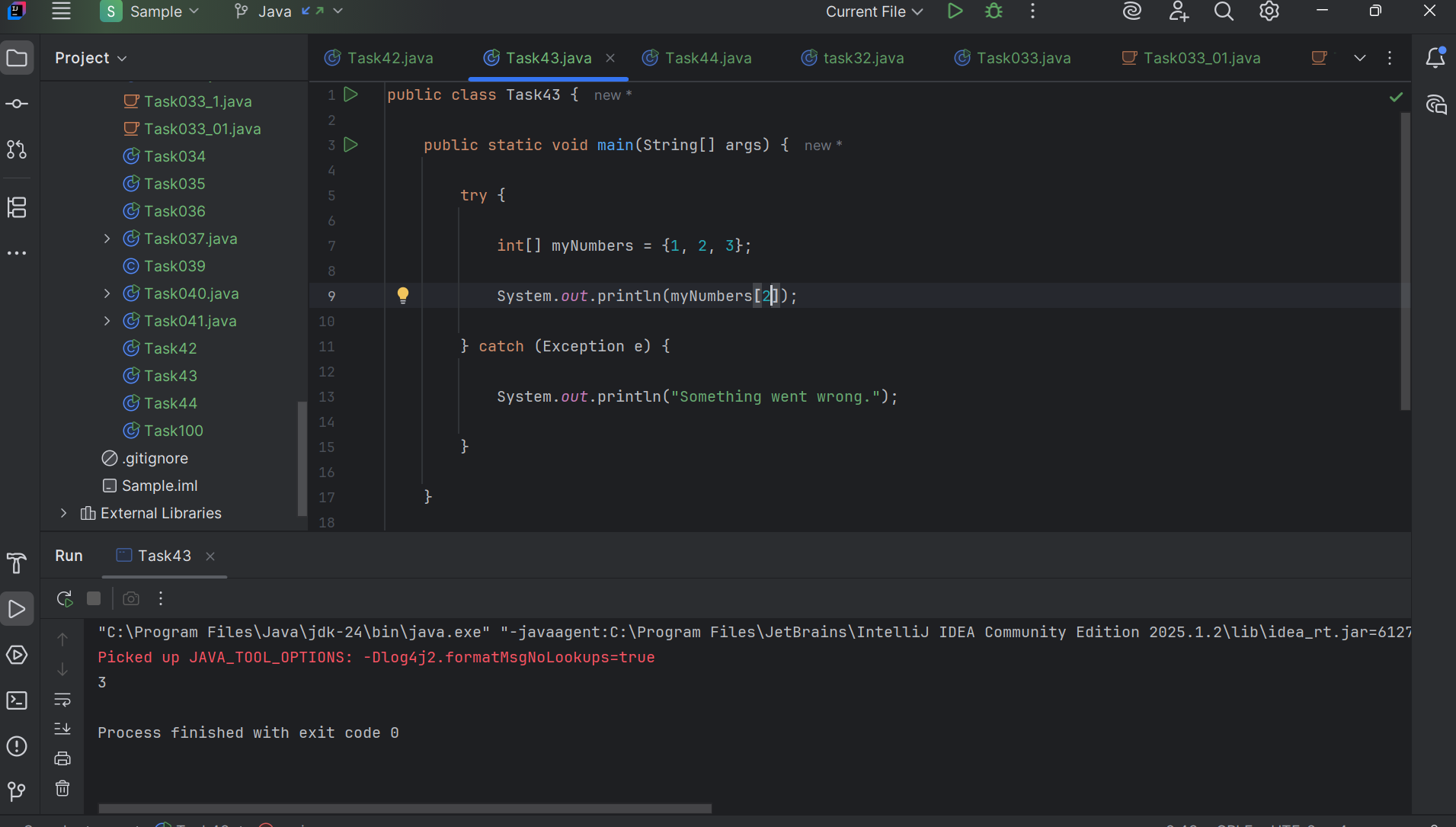


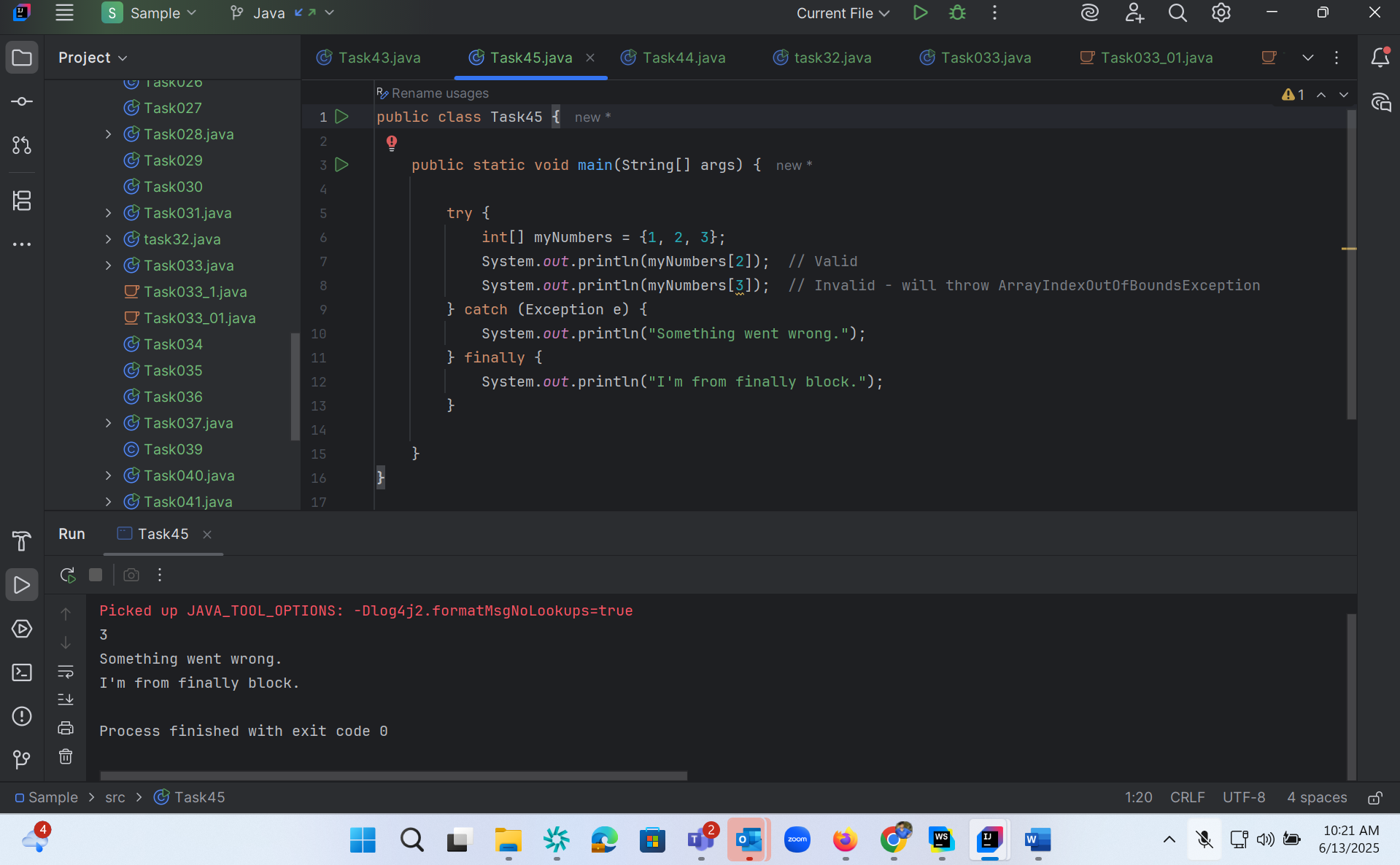
Task 3

public class Main {  
  public static void main(String[] args) {  
    try {  
      int[] myNumbers = {1, 2, 3};  
      System.out.println(myNumbers[10]);  
    } catch (Exception e) {  
      System.out.println("Something went wrong.");  
    } finally {  
      System.out.println("I'm from finally block.");  
    }  
  }  
}

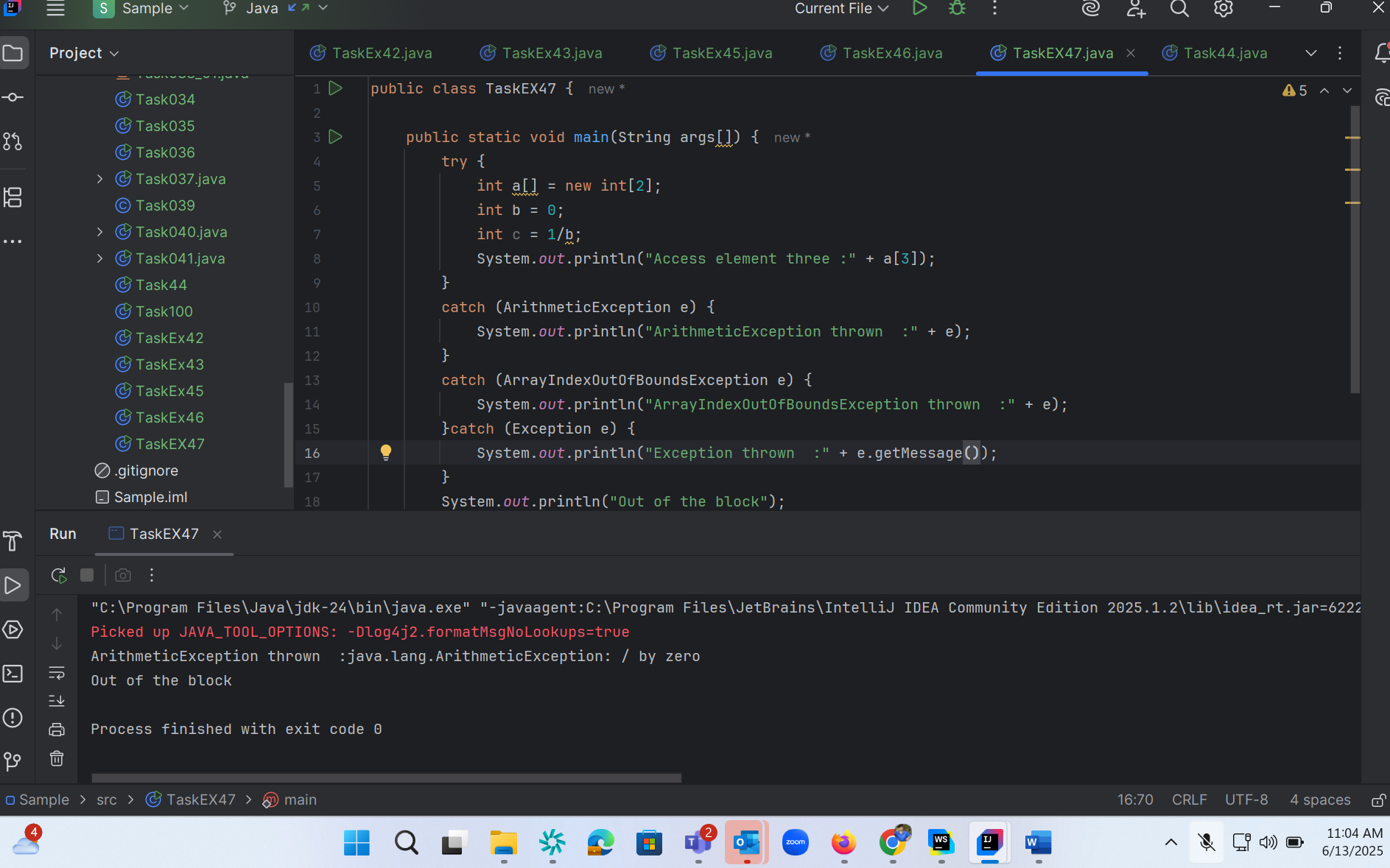


Task 4:

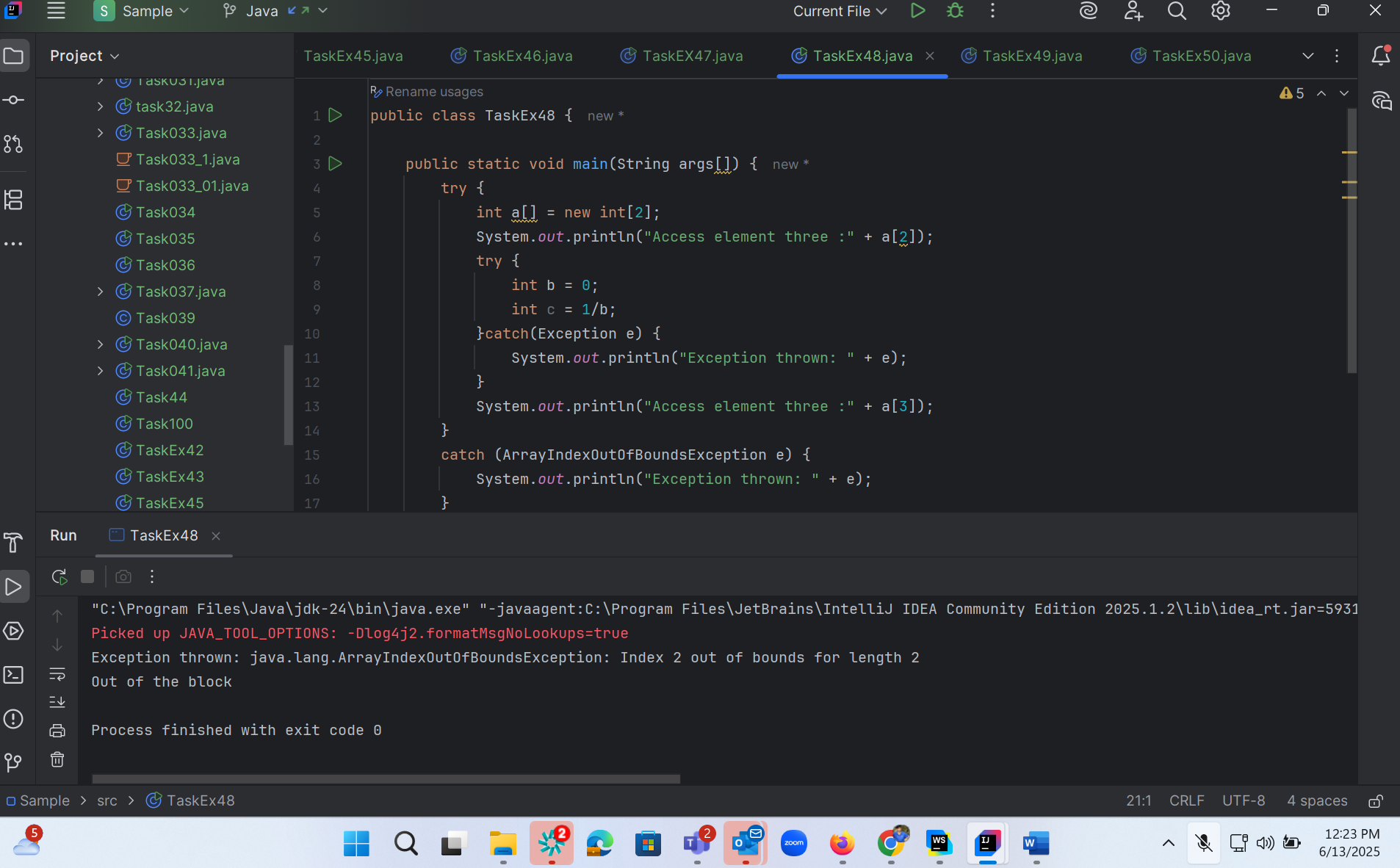




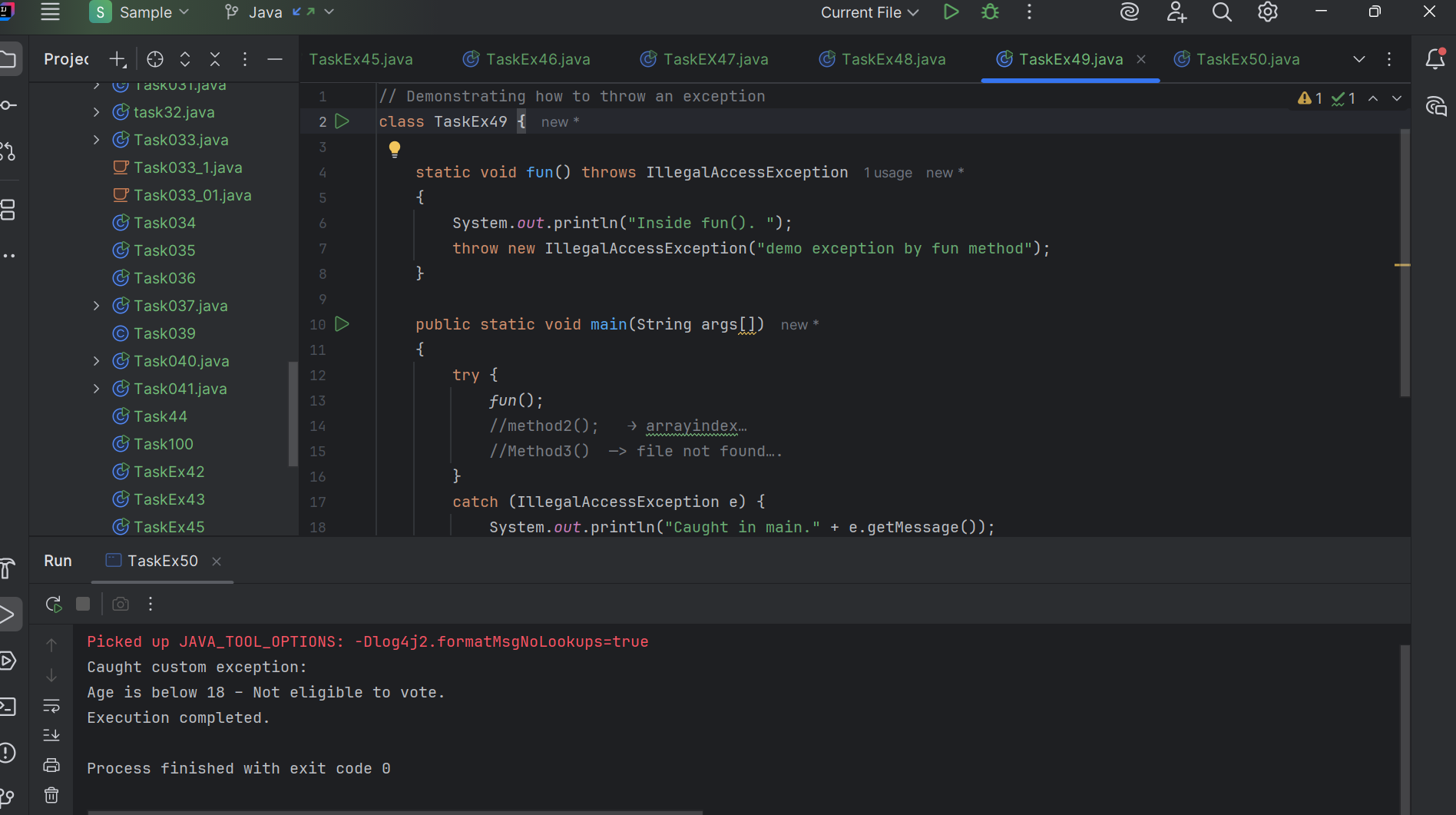
Task 5



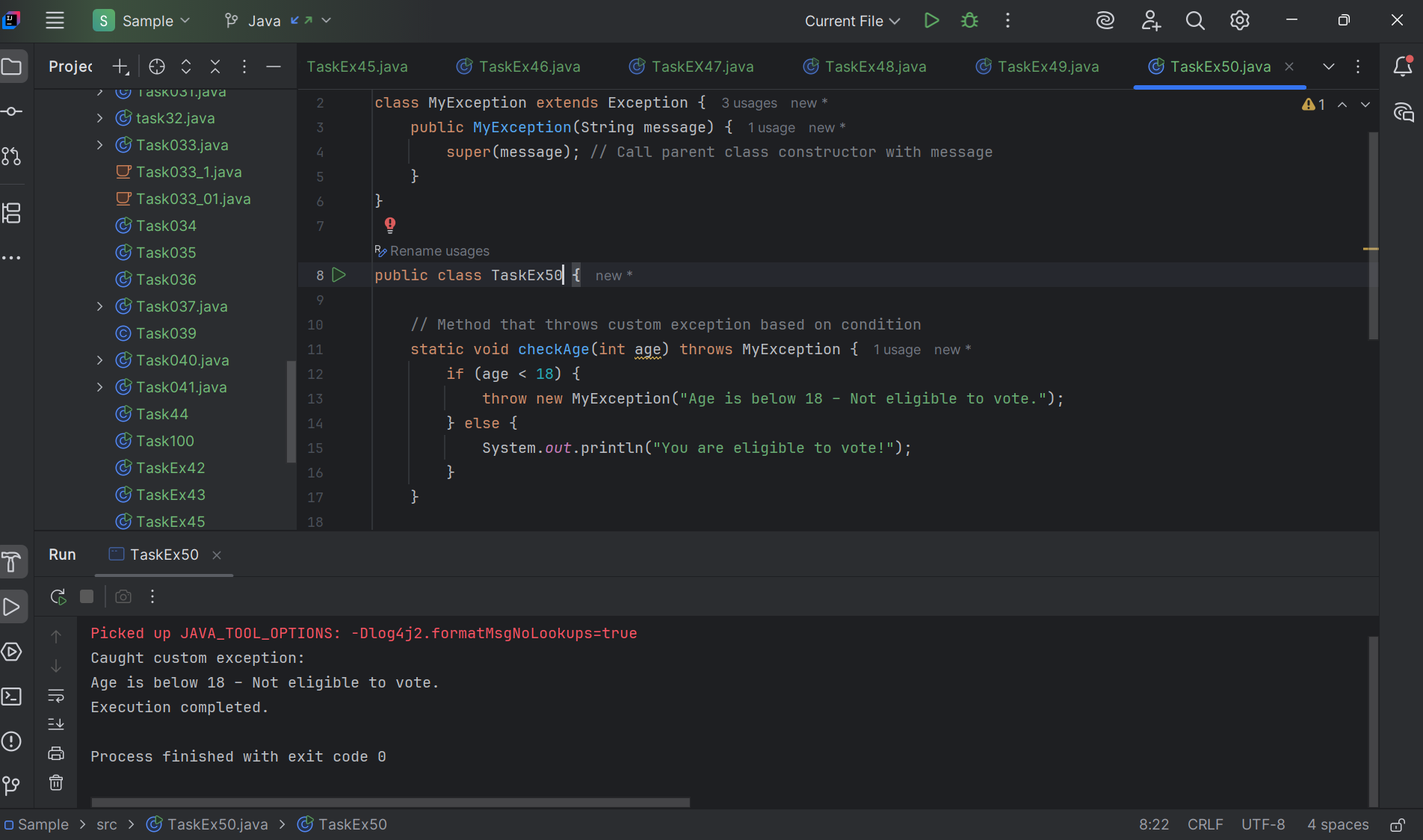
Task 6 :



Task 7:



Task : 8



Task 9:

import java.util.ArrayList;

class Main {

    public static void main (String[] args) {

       // Creating an ArrayList

       ArrayList<Integer> a = new ArrayList<Integer>();

       // Adding Element in ArrayList

       a.add(1);

       a.add(2);

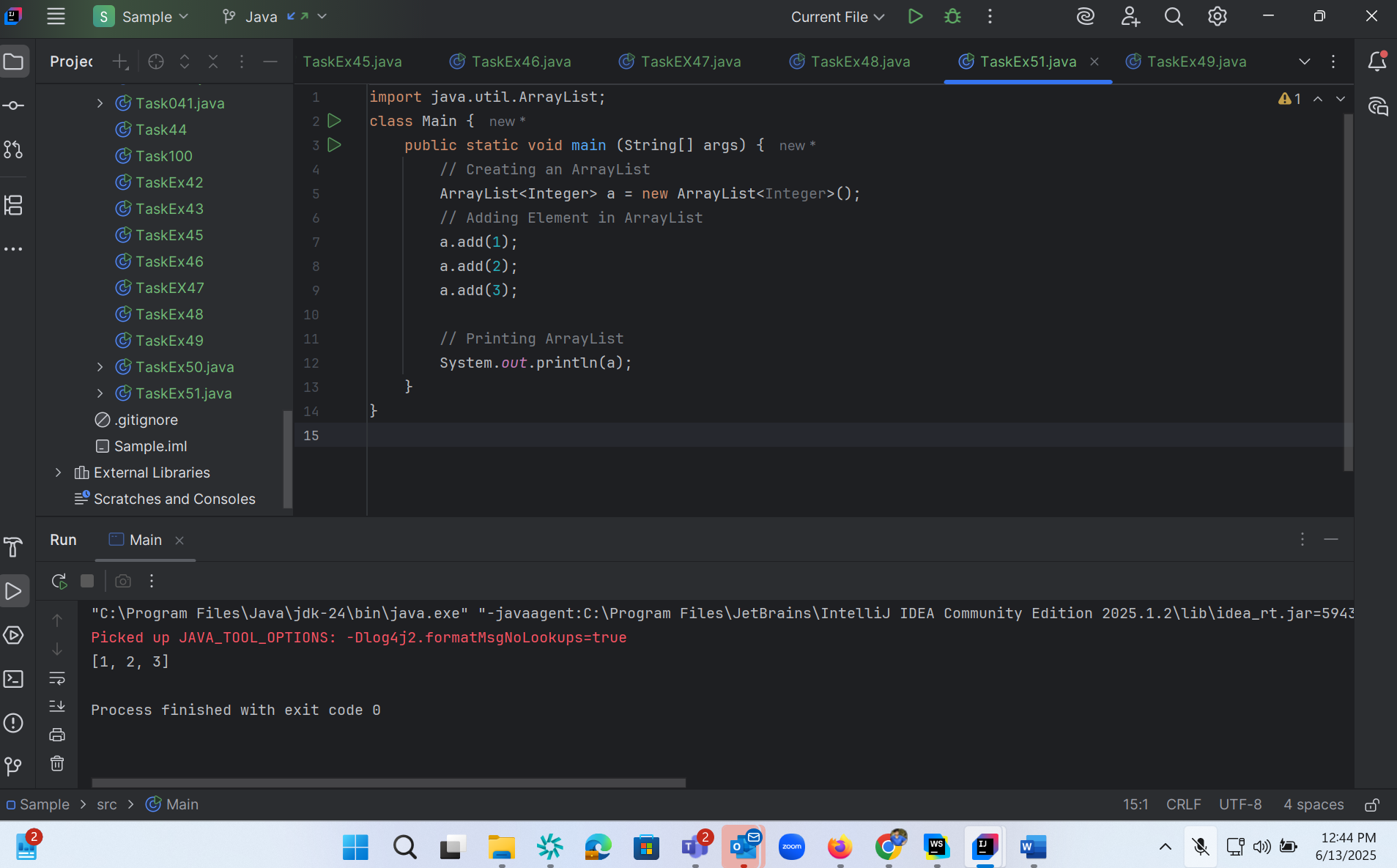
       a.add(3);

       // Printing ArrayList

       System.out.println(a);

    }

}



Task 10 :

// Addition, Deletion and Updation of Element

import java.util.\*;

class Main {

    public static void main(String args[]){

        ArrayList<String> al = new ArrayList<>();

        al.add("Prasunamba");

        al.add("Meher");

       System.out.println("Orignal List : "+al);

            al.add(1, "Hello");

       System.out.println("After Adding element at index 1 : "+ al);

              al.remove(0);

       System.out.println("Element removed from index 0 : "+ al);

       al.remove("Prasunamba");

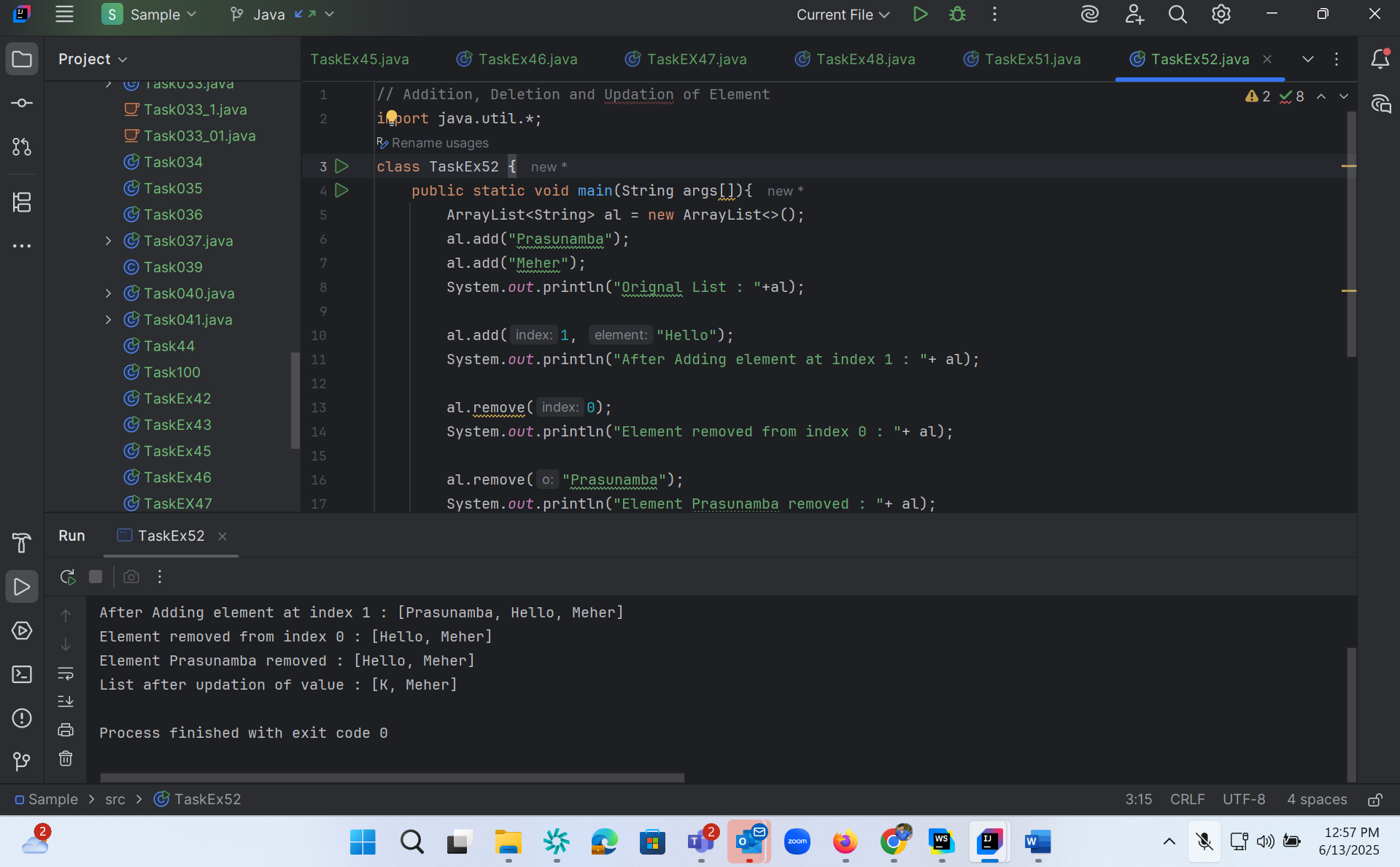
       System.out.println("Element Prasunamba removed : "+ al);

              al.set(0, "K");

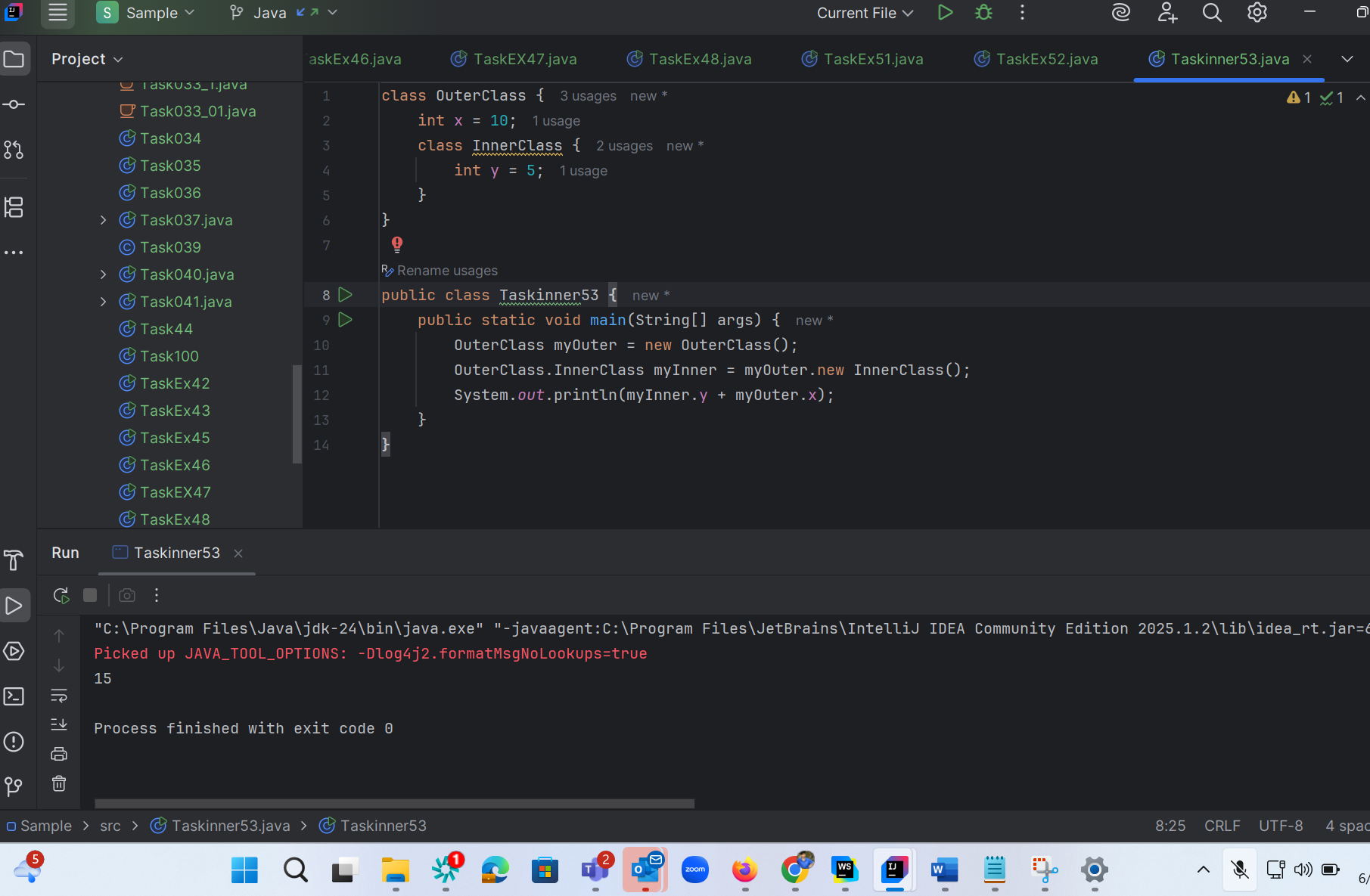
        System.out.println("List after updation of value : "+al);

    }

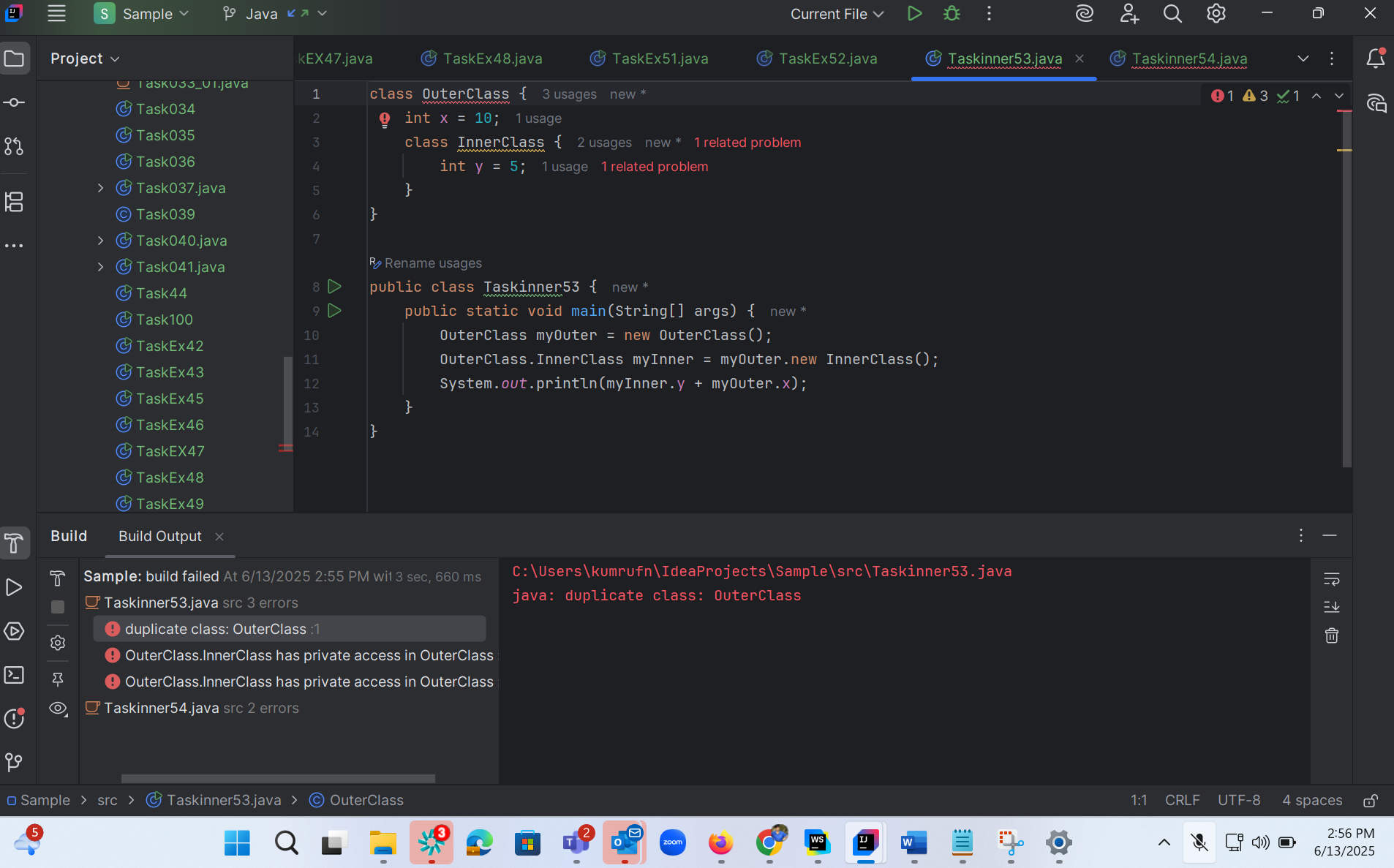
}



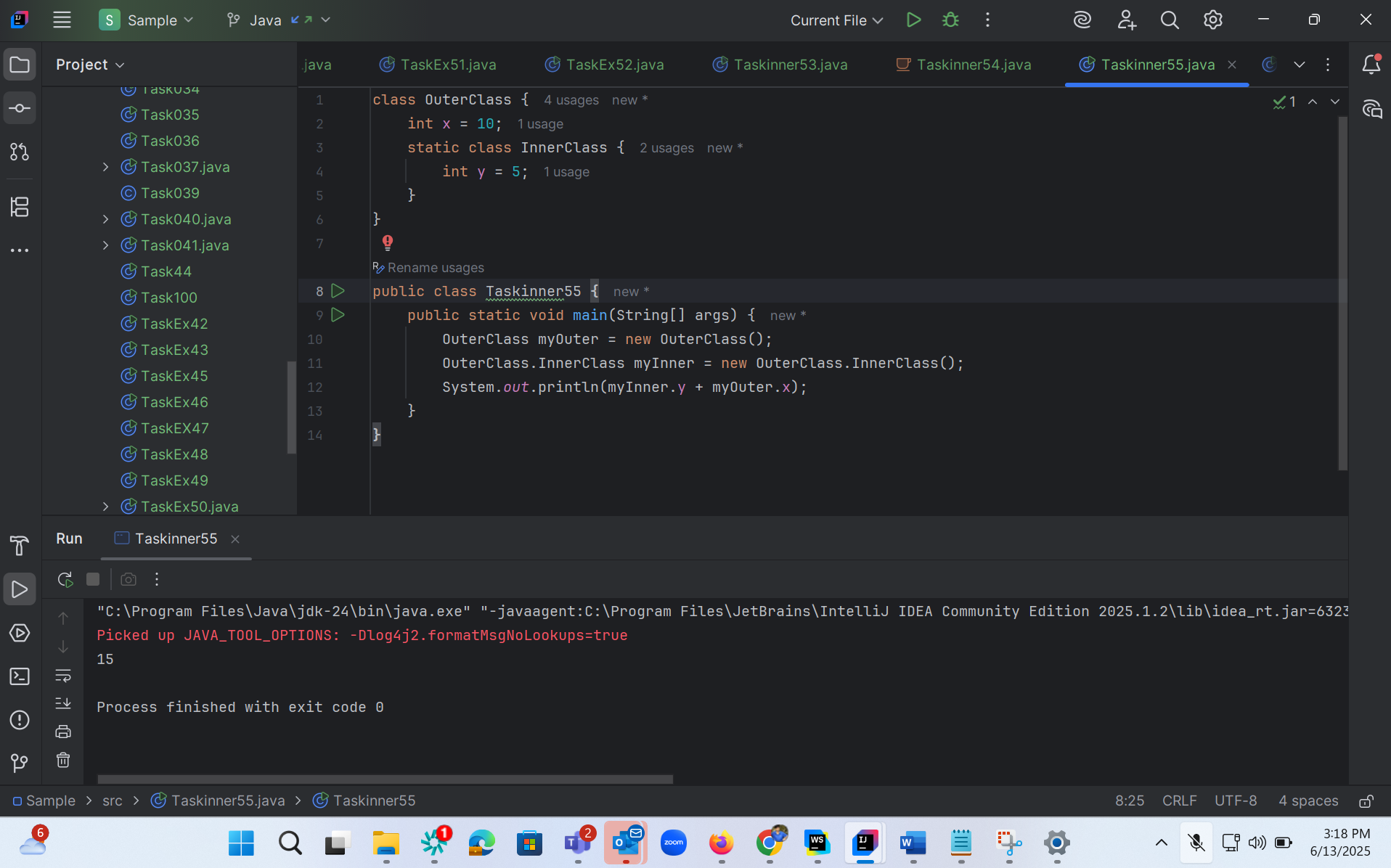
Task 11 :



Task 12:



Task 13:



Task 14:

Task 15 :

Task 16 :

Task 16:  ⇒  Home task..

Inheritance, Abstraction concepts..

Classes customer/ person , employee, Manager  … 2 variables in each class

Driver class – display all the variables… toString();

Hint : use getter and setters..

Person (parent)  —---> Employee (child) —--> Manager (sub Child)

Task 17:

What are the features of Java 8?

Java 8 Features:

Lambda Exp..

Interfaces can have body (default and static)

Foreach

Collection Api

Stream Api

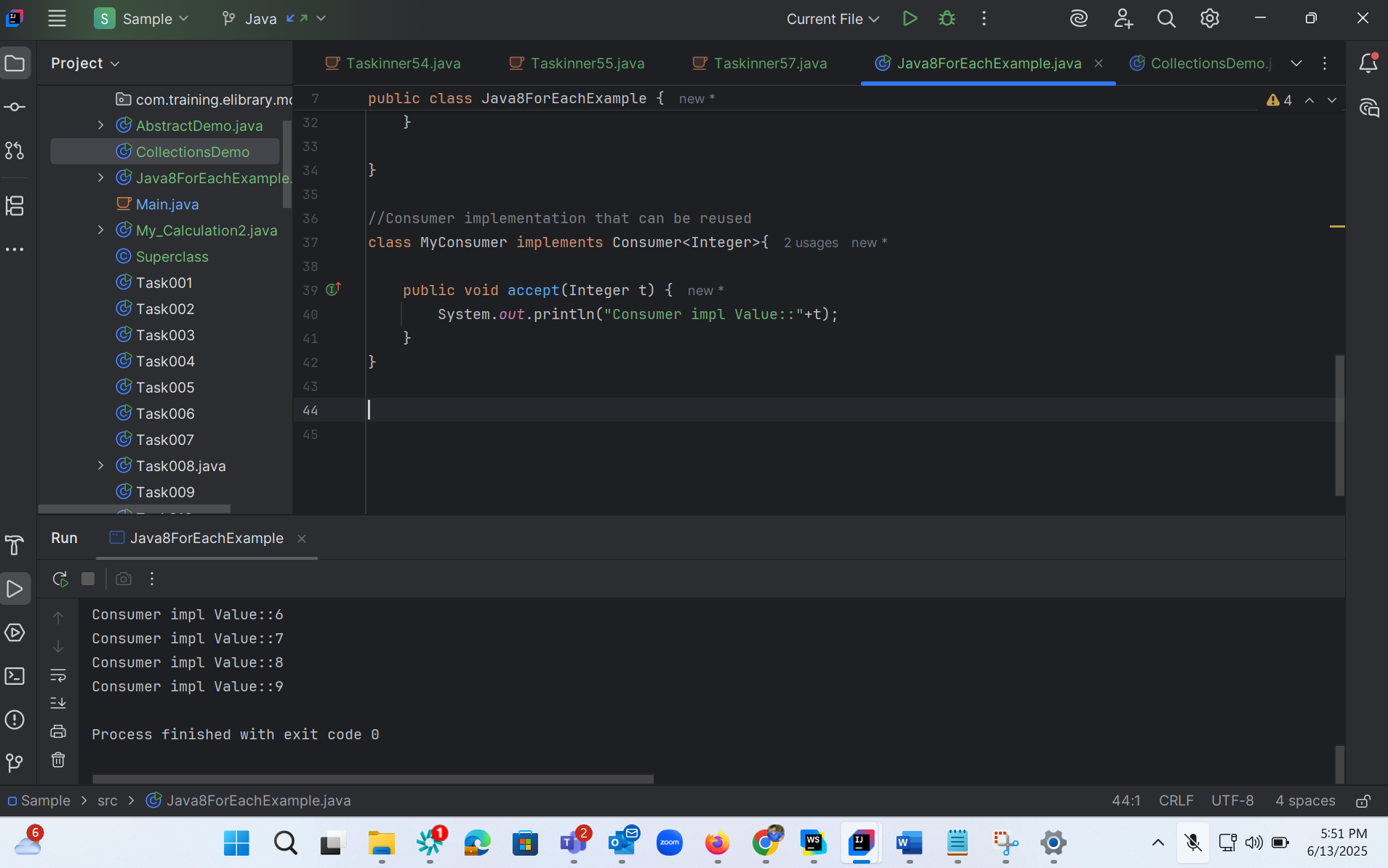
Java io improvements

Functional Interfaces

Method references

Time ApI

Task 18:



Task 19:

<https://www.digitalocean.com/community/tutorials/java-8-features-with-examples>

Plz go through this link for Java 8 features..

Task 20 :

