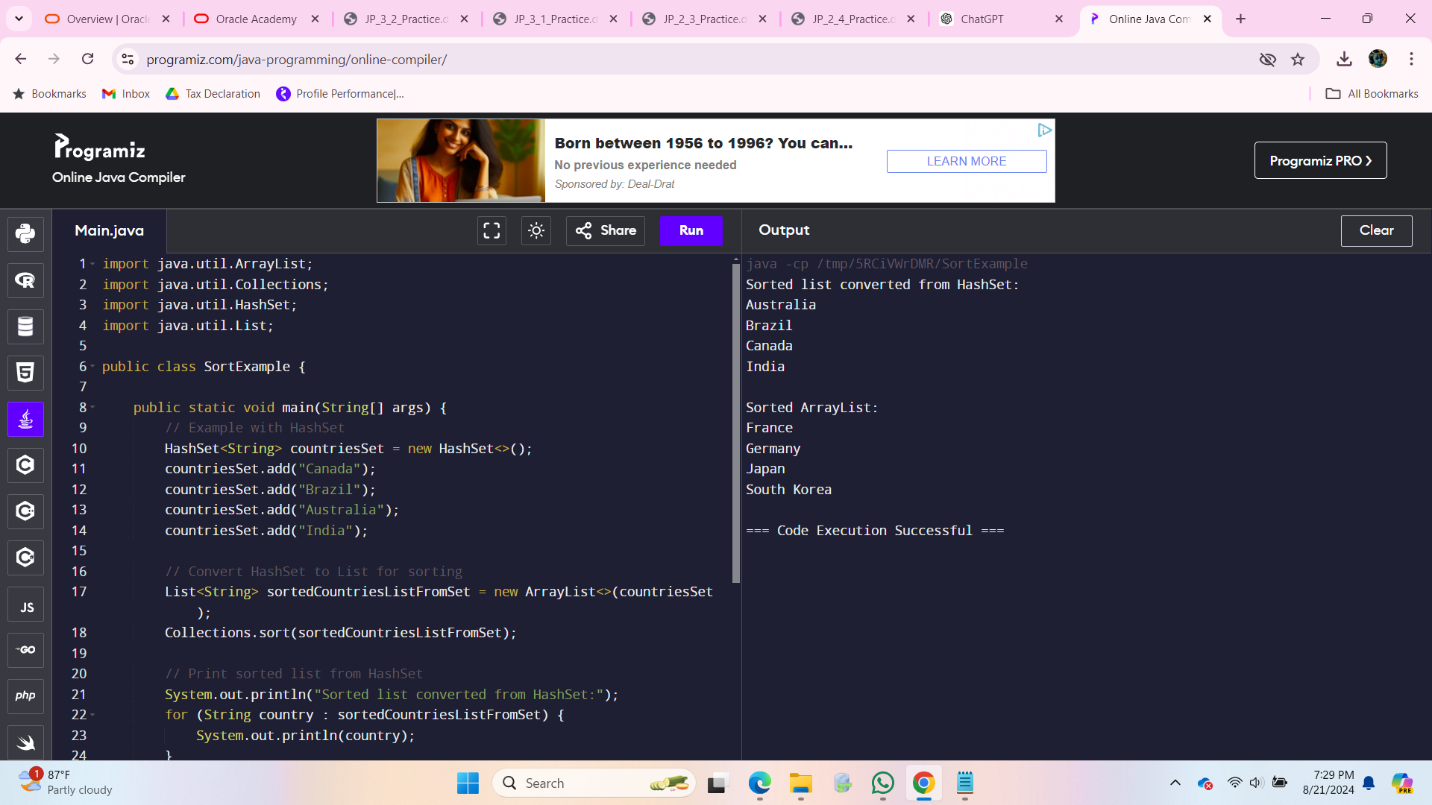
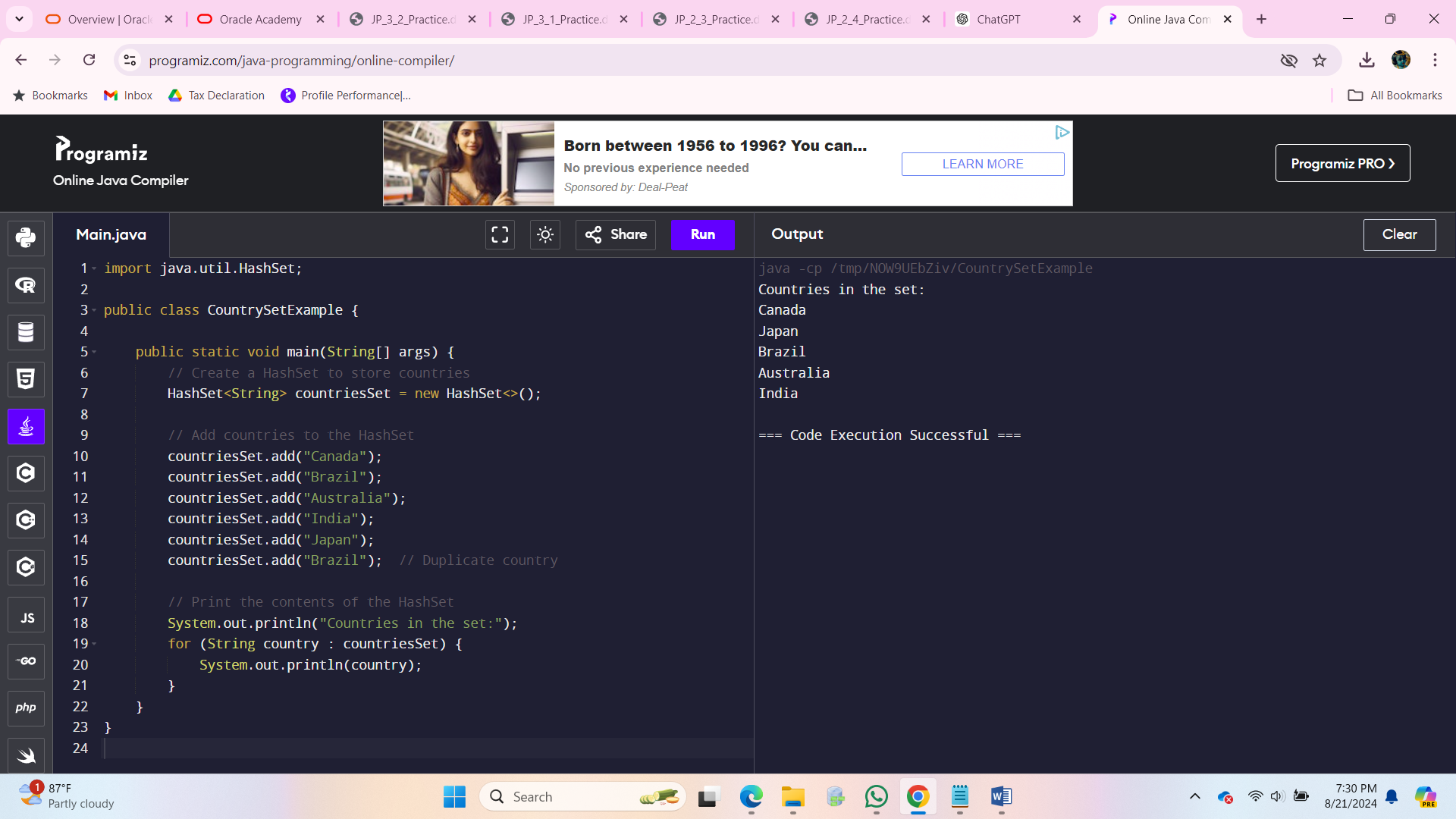
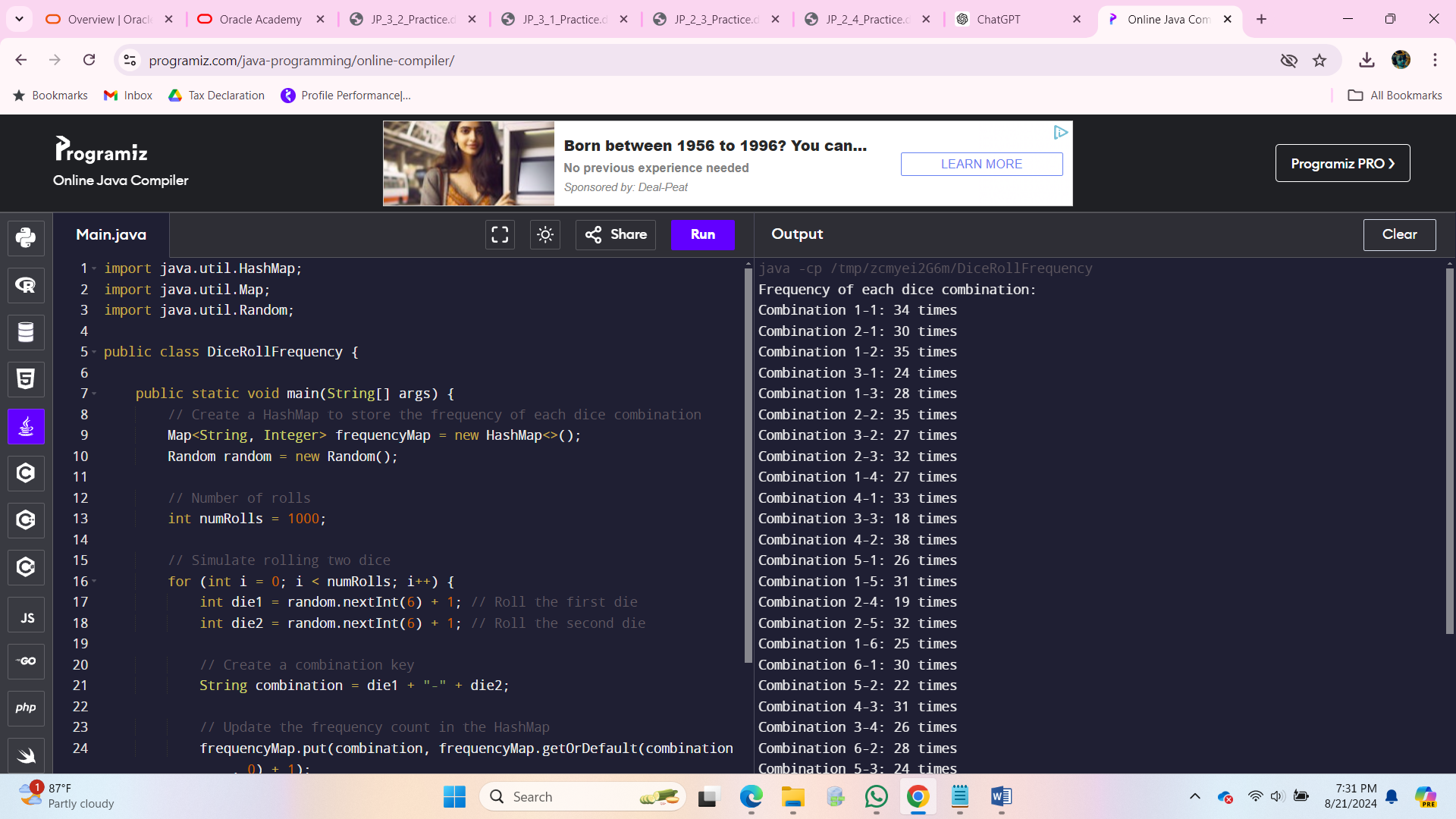
****

 **Set**:

* **Purpose**: To store unique elements without duplicates.
* **Usage**: Used when you need a collection that does not allow duplicate entries and where the order of elements is not a concern.
* **Examples**: HashSet, LinkedHashSet, TreeSet.

 **List**:

* **Purpose**: To store elements in a specific order, with the ability to access and modify elements by their index.
* **Usage**: Used when you need to maintain the order of elements and may require duplicates. Provides random access to elements.

**1.**

**a. Open the javabank.java File**

* Ensure you have the javabank.java file open in your IDE.

**b. Find and Comment Out the Static One-Dimensional Array Declaration**

* Locate the line that creates the static one-dimensional array for storing AbstractBankAccount objects. It might look something like this:
* static AbstractBankAccount myAccounts[] = new AbstractBankAccount[MAXACCOUNTS];

#### Update Array Operations to Work with ArrayList

You will need to replace all operations that use the array with the corresponding ArrayList methods. Here are some common conversions:

1. **Adding Accounts to ArrayList**:

**Before**:

static AbstractBankAccount myAccounts[] = new AbstractBankAccount[MAXACCOUNTS];

**Removing Accounts from ArrayList** (if applicable):

myAccounts[i] = null; // Or similar approach

**Accessing the Size of the ArrayList**:

**Before**:

int numberOfAccounts = myAccounts.length;

#### Identify and Fix Errors

Compile the code to identify any errors that arise from the switch to ArrayList. The most common errors will likely be related to replacing array indexing with ArrayList methods.

For example, if you had any code that relied on direct indexing of the array, such as:

java

myAccounts[i].getBalance();

#### Run and Test the Java Application

Once you have updated the code:

1. **Create Multiple Accounts**: Test by creating instances of different types of accounts (e.g., CreditAccount, SavingsAccount) and adding them to the ArrayList.
2. **Display Accounts**: Make sure the accounts are correctly displayed in the textArea or any other UI component you are using.

Here’s a simple example to create and display accounts:

// Example code to add and display accounts

public class JavaBankApp {

public static void main(String[] args) {

// Create and add accounts

myAccounts.add(new CreditAccount("John Doe", 1001, 5000));

myAccounts.add(new SavingsAccount("Jane Smith", 1002, 3000));

// Display accounts (assuming a textArea is available for display)

StringBuilder displayText = new StringBuilder();

for (AbstractBankAccount account : myAccounts) {

displayText.append(account.toString()).append("\n");

}

// Display in textArea

textArea.setText(displayText.toString());

}

}