Q-1)  Write a Java program that reads a string from the user and uses StringTokenizer to split the string into individual words. Print each word on a new line.

**Program:**

**package** Home;

**import** java.util.Scanner;

**import** java.util.StringTokenizer;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

// Create a Scanner object to read input from the user

Scanner sc = **new** Scanner(System.***in***);

// Prompt the user to enter a string

System.***out***.print("Enter a string: ");

String inputString = sc.nextLine();

// Create a StringTokenizer object to tokenize the input string

StringTokenizer tokenizer = **new** StringTokenizer(inputString);

// Iterate through the tokens and print each word on a new line

System.***out***.println("Words in the string:");

**while** (tokenizer.hasMoreTokens()) {

System.***out***.println(tokenizer.nextToken());

}

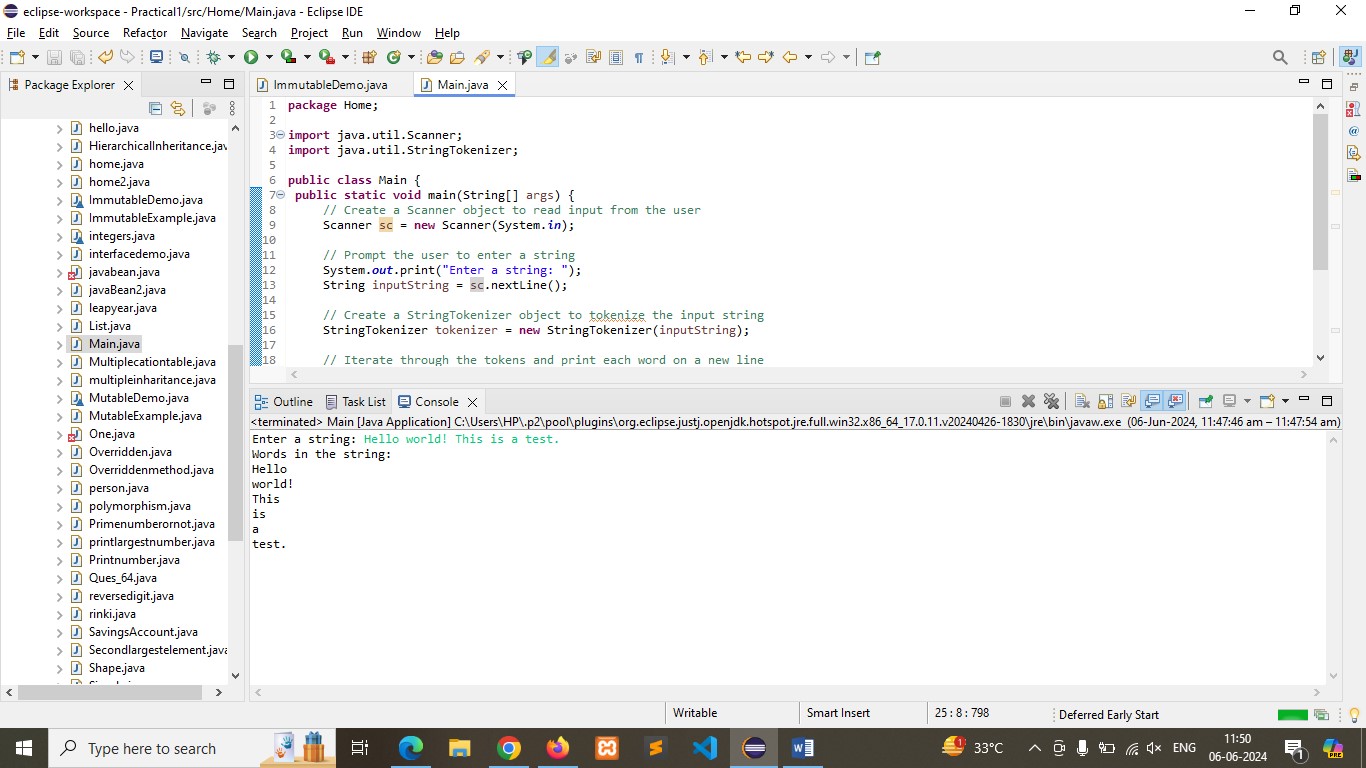
// Close the scanner

sc.close();

}

}

Output:



Q-2)    Write a Java program that reads a string from the user and uses StringTokenizer to count the number of words in the string.

**Program:**

**package** Home;

**import** java.util.Scanner;

**import** java.util.StringTokenizer;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

// Create a Scanner object to read input from the user

Scanner sc = **new** Scanner(System.***in***);

// Prompt the user to enter a string

System.***out***.print("Enter a string: ");

String inputString = sc.nextLine();

// Create a StringTokenizer object to tokenize the input string

StringTokenizer tokenizer = **new** StringTokenizer(inputString);

// Count the number of words in the string

**int** wordCount = tokenizer.countTokens();

// Print the number of words in the string

System.***out***.println("Number of words in the string: " + wordCount);

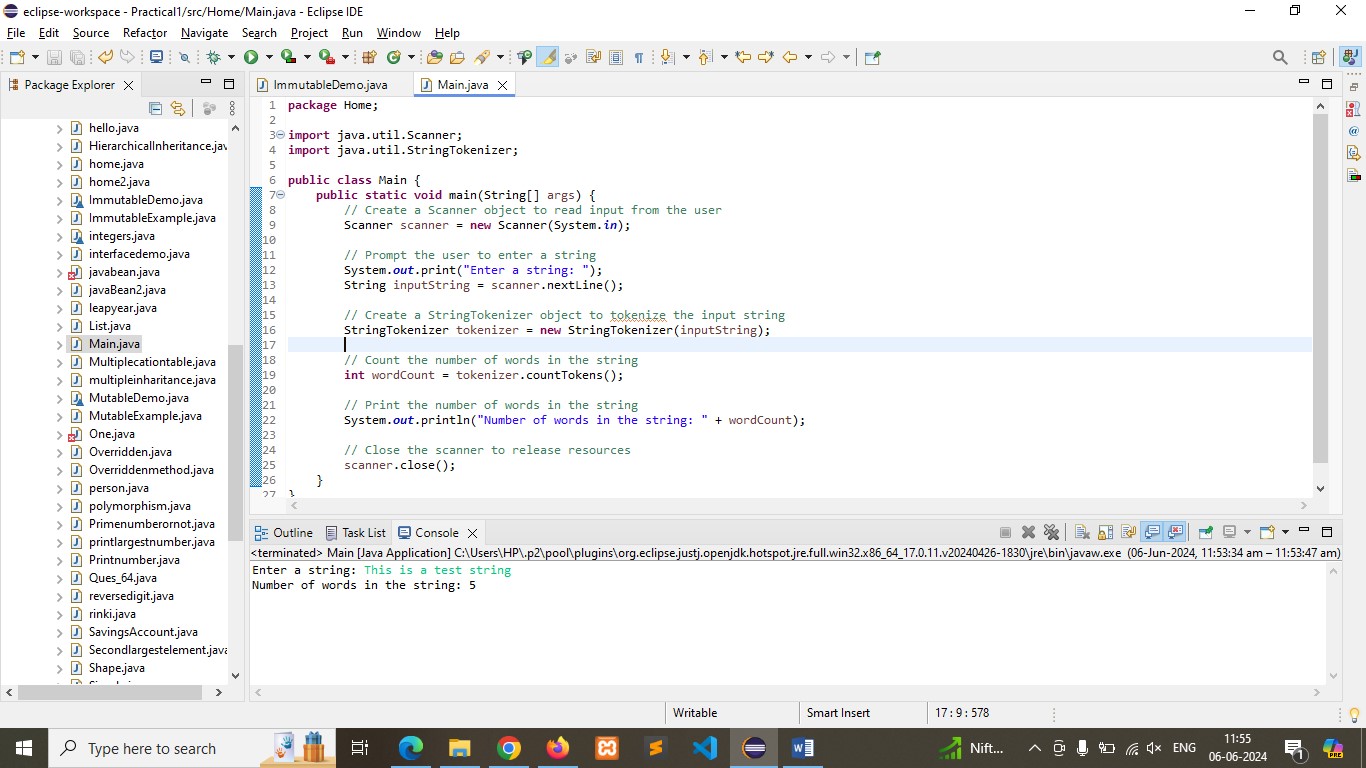
// Close the scanner to release resources

sc.close();

}

}

**Output:**



Q-3) Write a Java program to create a LinkedList of strings, add elements at specific positions (beginning, middle, end), and print the list.

**Program:**

**package** Home;

**import** java.util.LinkedList;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

// Create a LinkedList to store strings

LinkedList<String> linkedList = **new** LinkedList<>();

// Add elements at specific positions

// Add element at the beginning

linkedList.addFirst("beginning");

// Add element in the middle

linkedList.add(linkedList.size() / 2, "Middle");

// Add element at the last

linkedList.addLast("end");

// Print the LinkedList

System.***out***.println("LinkedList elements:");

**for** (String element : linkedList) {

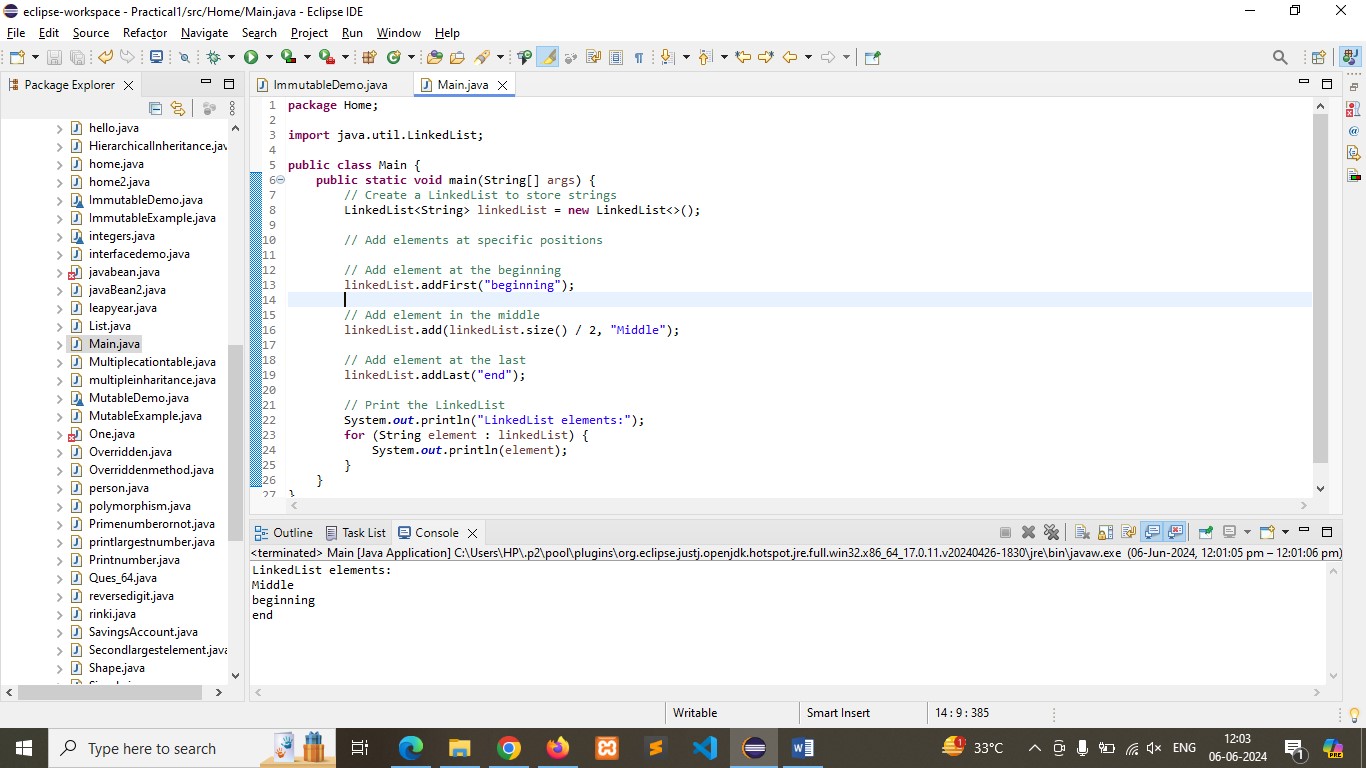
System.***out***.println(element);

}

}

}

**Output**:



Q-4)   Write a Java program to sort a given array list.

**Program:**

**package** Home;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

// Create an ArrayList of integers

ArrayList<Integer> arrayList = **new** ArrayList<>();

// Add elements to the ArrayList

arrayList.add(5);

arrayList.add(2);

arrayList.add(7);

arrayList.add(1);

arrayList.add(9);

// Print the original ArrayList

System.***out***.println("Original ArrayList:");

System.***out***.println(arrayList);

// Sort the ArrayList

Collections.*sort*(arrayList);

// Print the sorted ArrayList

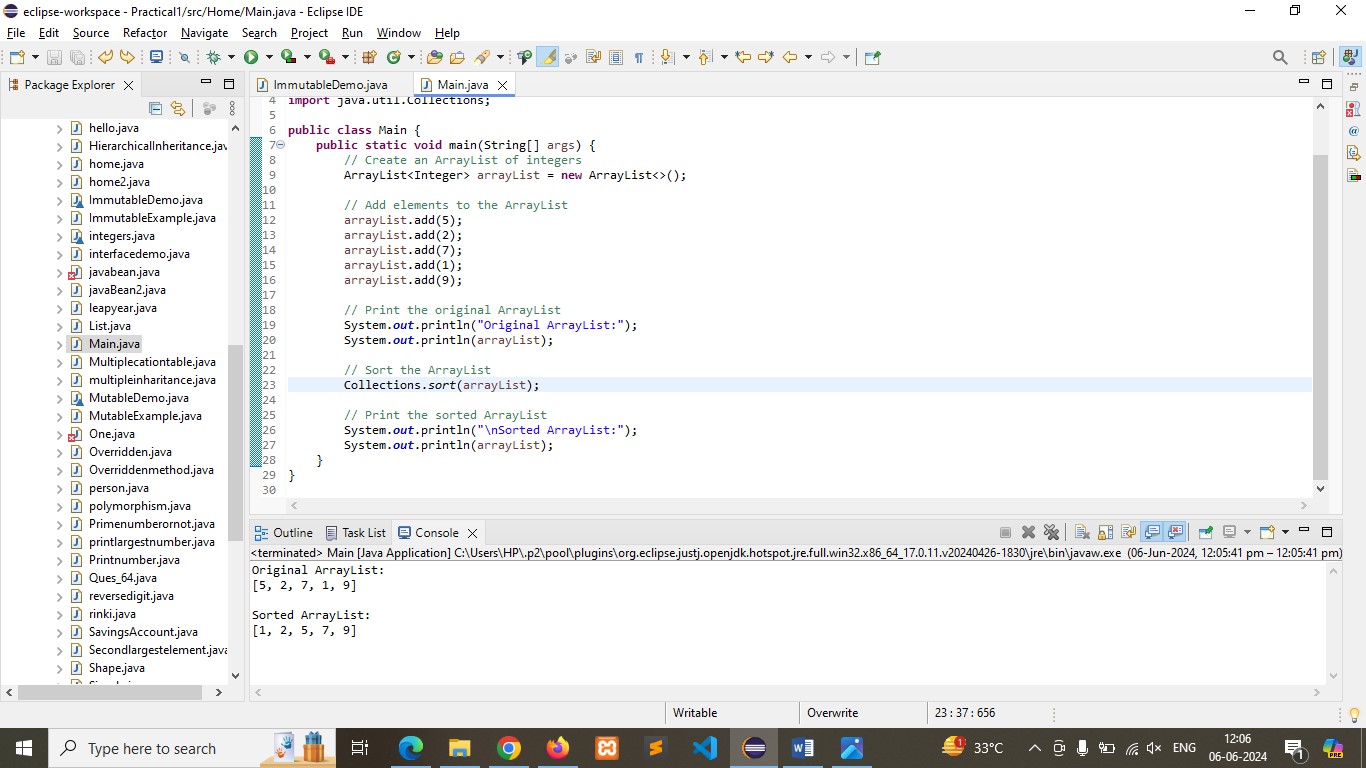
System.***out***.println("\nSorted ArrayList:");

System.***out***.println(arrayList);

}

}

**Output:**



Q**-**5)    Write a Java program to replace the second element of an ArrayList with the specified element.

**Program**:

**package** Home;

**import** java.util.ArrayList;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

// Create an ArrayList of strings

ArrayList<String> arrayList = **new** ArrayList<>();

// Add elements to the ArrayList

arrayList.add("First");

arrayList.add("Second");

arrayList.add("Third");

arrayList.add("Fourth");

// Print the original ArrayList

System.***out***.println("Original ArrayList:");

System.***out***.println(arrayList);

// Replace the second element with the specified element

String specifiedElement = "New Second";

arrayList.set(1, specifiedElement);

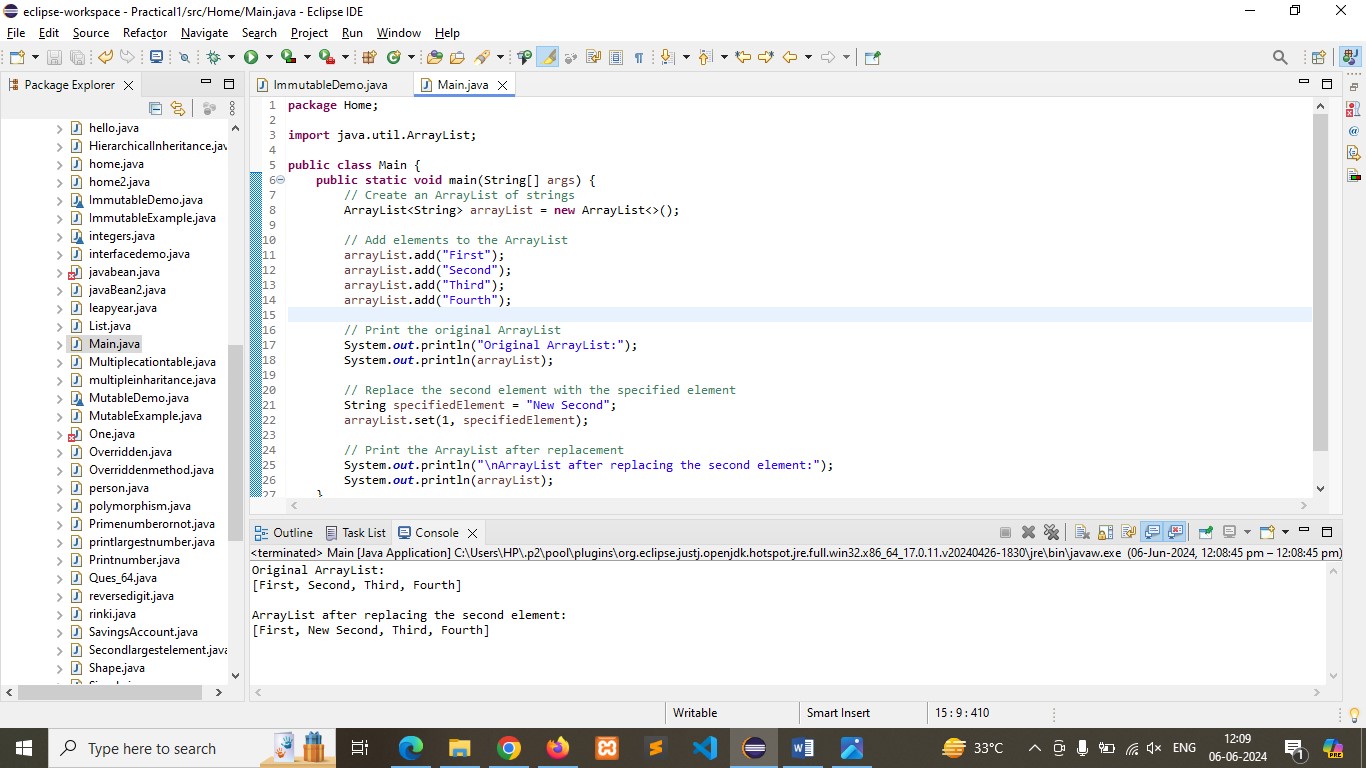
// Print the ArrayList after replacement

System.***out***.println("\nArrayList after replacing the second element:");

System.***out***.println(arrayList);

}

}

**Output:** 

Q-6)     Write a Java program to iterate a linked list in reverse order.

**Program**

:

**package** Home;

**import** java.util.LinkedList;

**import** java.util.ListIterator;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

// Create a LinkedList of strings

LinkedList<String> linkedList = **new** LinkedList<>();

// Add elements to the LinkedList

linkedList.add("First");

linkedList.add("Second");

linkedList.add("Third");

linkedList.add("Fourth");

// Print the original LinkedList

System.***out***.println("Original LinkedList:");

System.***out***.println(linkedList);

// Create a ListIterator to iterate over the LinkedList in reverse order

ListIterator<String> iterator = linkedList.listIterator(linkedList.size());

// Iterate over the LinkedList in reverse order and print each element

System.***out***.println("\nLinkedList elements in reverse order:");

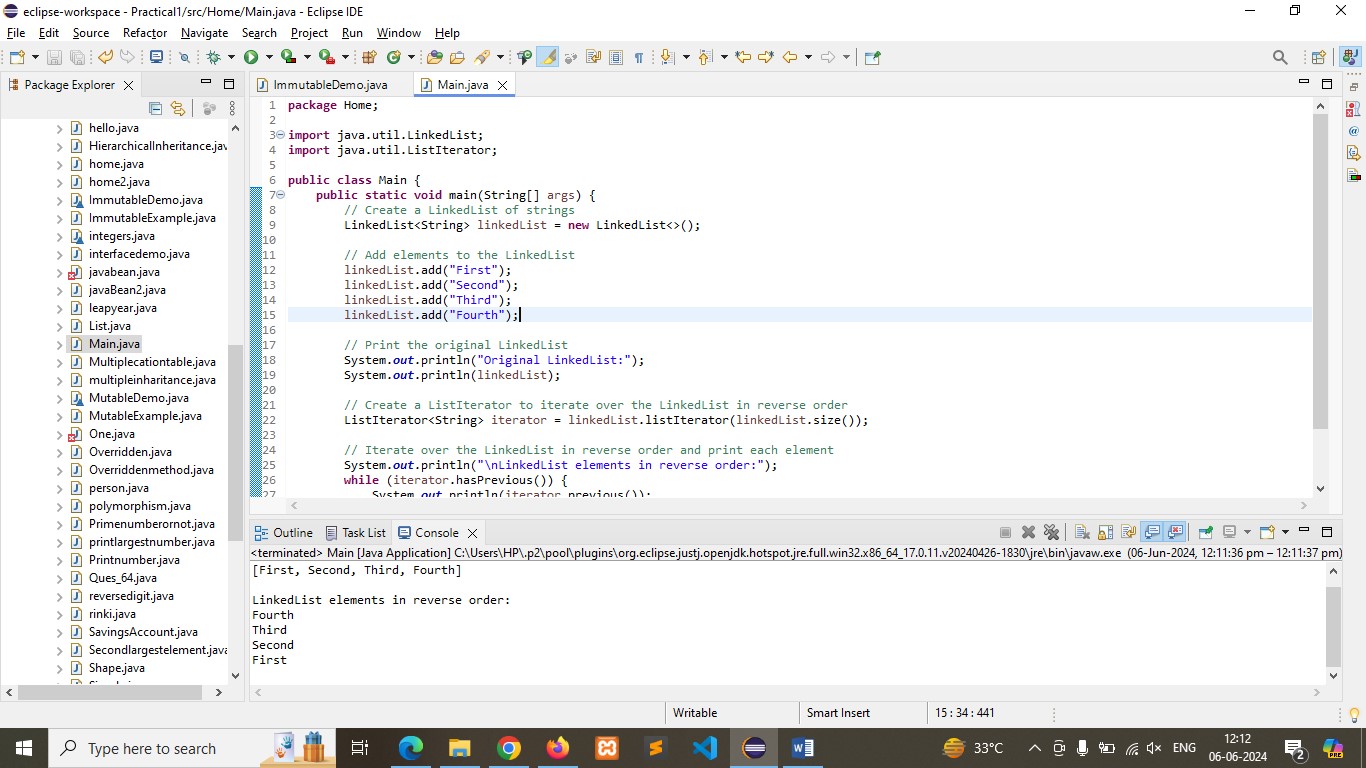
**while** (iterator.hasPrevious()) {

System.***out***.println(iterator.previous());

}

}

}

**Output:** 

Q**-**7.     Write a Java program to retrieve, but not remove, the last element of a linked list.

**Program:**

**package** Home;

**import** java.util.LinkedList;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

// Create a LinkedList of integers

LinkedList<Integer> linkedList = **new** LinkedList<>();

// Add elements to the LinkedList

linkedList.add(1);

linkedList.add(2);

linkedList.add(3);

linkedList.add(4);

// Print the original LinkedList

System.***out***.println("Original LinkedList:");

System.***out***.println(linkedList);

// Retrieve the last element of the LinkedList

Integer lastElement = linkedList.getLast();

// Print the last element without removing it

System.***out***.println("\nLast element of the LinkedList (without removing):");

System.***out***.println(lastElement);

// Print the LinkedList to confirm that the last element is still present

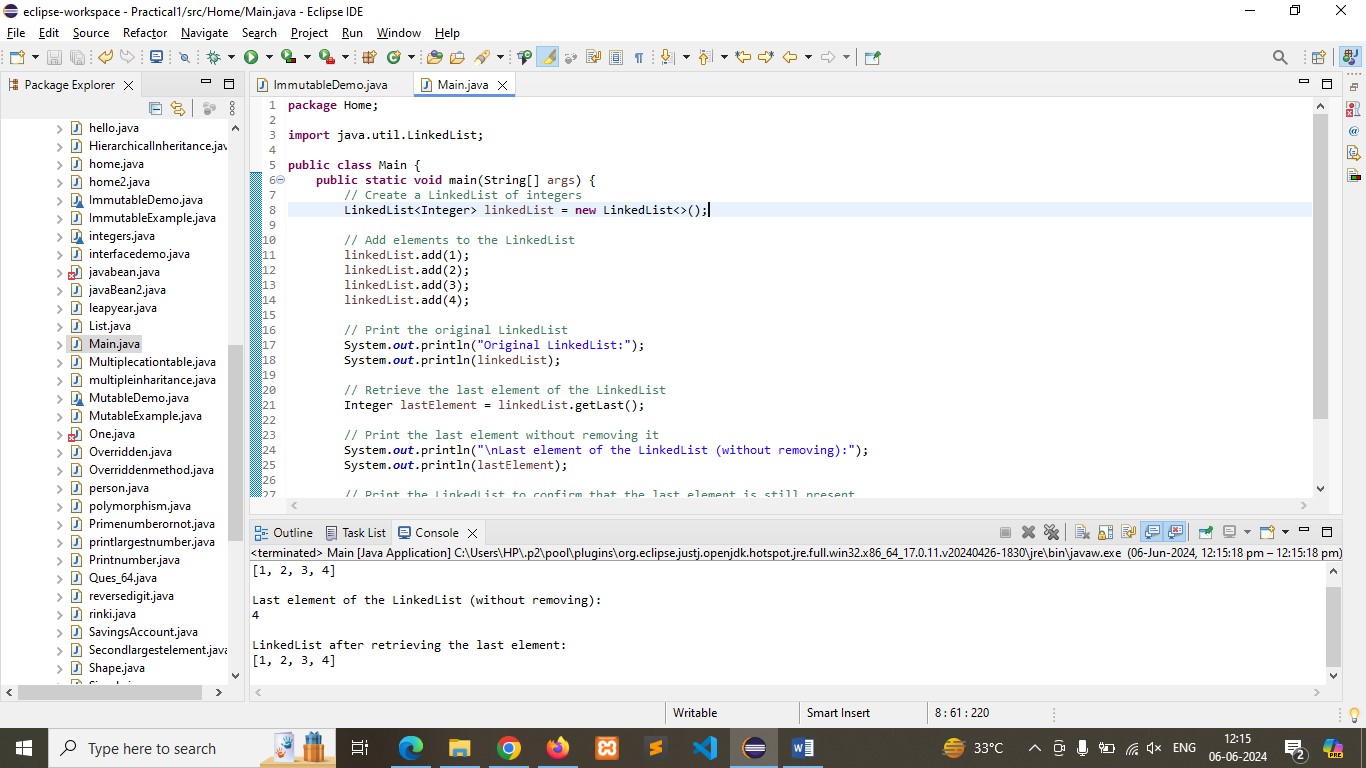
System.***out***.println("\nLinkedList after retrieving the last element:");

System.***out***.println(linkedList);

}

}

**Output:**



Q**-**8.     Write a Java program to create a LinkedList of integers and print all the elements.

**Program:**

**package** Home;

**import** java.util.LinkedList;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

// Create a LinkedList of integers

LinkedList<Integer> linkedList = **new** LinkedList<>();

// Add elements to the LinkedList

linkedList.add(1);

linkedList.add(2);

linkedList.add(3);

linkedList.add(4);

// Print all the elements in the LinkedList

System.***out***.println("LinkedList elements:");

**for** (Integer element : linkedList) {

System.***out***.println(element);

}

}

}

**Output**:

