1. Write the same programme to sort the integers 8, 4, 3,5,6 and the alphabetical string C, O, I, P, U, in ascending order. Show the resulting output.

CODE :

**package** lab6;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** Sort {

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

// Sorting integers

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

numbers.add(8);

numbers.add(4);

numbers.add(3);

numbers.add(5);

numbers.add(6);

System.***out***.println("Original list of integers: " + numbers);

// Sorting in ascending order

Collections.*sort*(numbers);

System.***out***.println("Sorted list of integers in ascending order: " + numbers);

// Sorting alphabetical strings

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

strings.add("C");

strings.add("O");

strings.add("I");

strings.add("P");

strings.add("U");

System.***out***.println("Original list of strings: " + strings);

// Sorting in ascending order

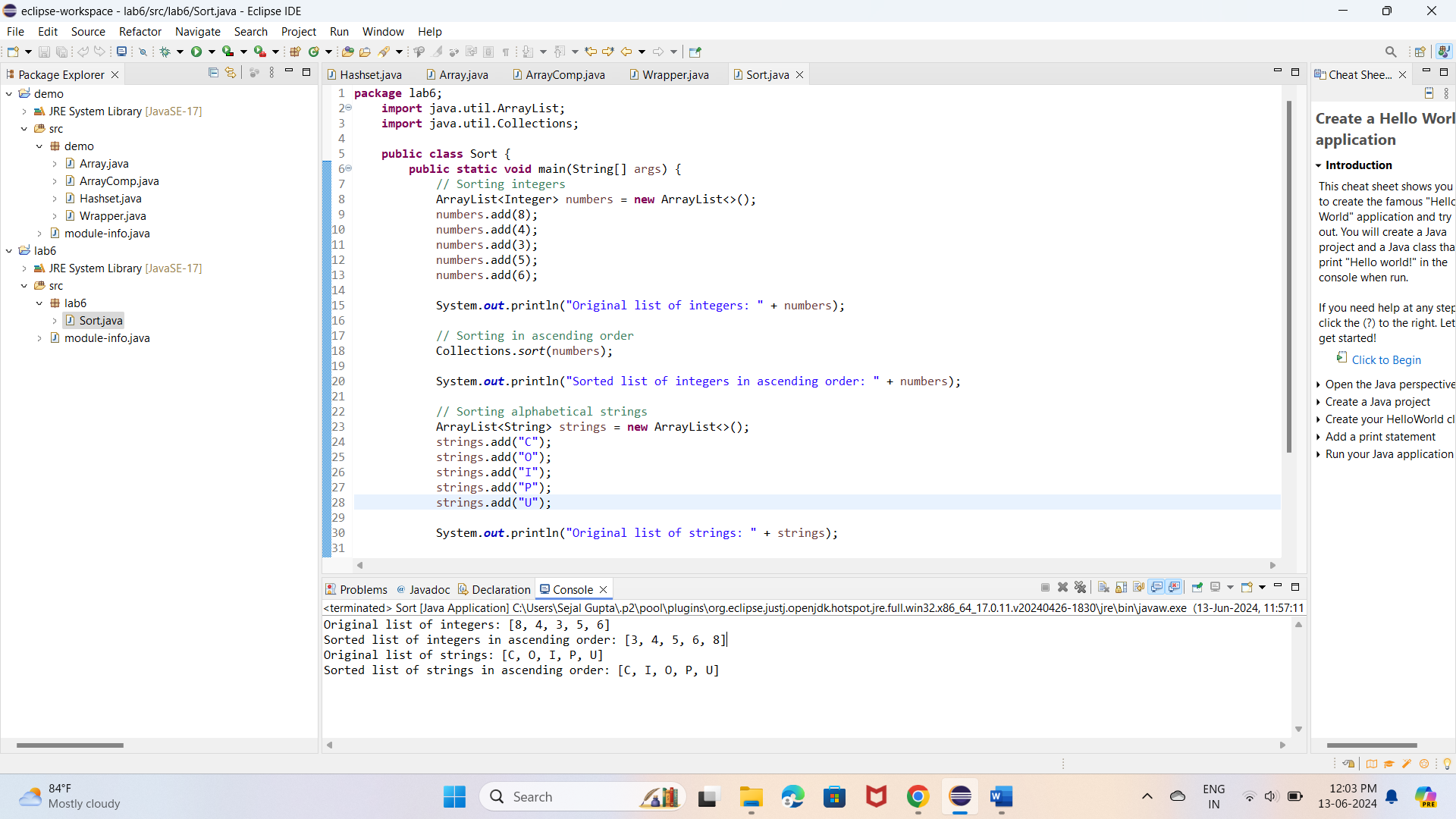
Collections.*sort*(strings);

System.***out***.println("Sorted list of strings in ascending order: " + strings);

}

}

Program-



2.Write a Java program to implement the bubble sort algorithm to sort an array of integers in ascending order.

CODE :

**package** lab6;

**public** **class** BubbleSort {

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

**int**[] arr = {8, 4, 3, 5, 6};

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

*printArray*(arr);

*bubbleSort*(arr);

System.***out***.println("\nSorted array in ascending order:");

*printArray*(arr);

}

// Bubble Sort function

**public** **static** **void** bubbleSort(**int**[] arr) {

**int** n = arr.length;

**boolean** swapped;

**for** (**int** i = 0; i < n - 1; i++) {

swapped = **false**;

**for** (**int** j = 0; j < n - i - 1; j++) {

**if** (arr[j] > arr[j + 1]) {

// Swap arr[j] and arr[j+1]

**int** temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

swapped = **true**;

}

}

// If no two elements were swapped in the inner loop, then break

**if** (!swapped) {

**break**;

}

}

}

// Utility function to print an array

**public** **static** **void** printArray(**int**[] arr) {

**for** (**int** num : arr) {

System.***out***.print(num + " ");

}

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

}

}

Program :

A screenshot of a computer

Description automatically generated

3.  Write a program to input an array 10 elements and print the cube of prime numbers in it.

CODE :

**package** lab6;

**import** java.util.Scanner;

**public** **class** PrimeCubes {

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

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

**int**[] arr = **new** **int**[10];

// Input 10 elements into the array

System.***out***.println("Enter 10 integers:");

**for** (**int** i = 0; i < 10; i++) {

arr[i] = scanner.nextInt();

}

// Find prime numbers and print their cubes

System.***out***.println("\nCubes of prime numbers in the array:");

**for** (**int** num : arr) {

**if** (*isPrime*(num)) {

**long** cube = (**long**) num \* num \* num; // Compute cube

System.***out***.println("Cube of " + num + " = " + cube);

}

}

scanner.close();

}

// Function to check if a number is prime

**public** **static** **boolean** isPrime(**int** num) {

**if** (num <= 1) {

**return** **false**;

}

**if** (num <= 3) {

**return** **true**;

}

**if** (num % 2 == 0 || num % 3 == 0) {

**return** **false**;

}

**for** (**int** i = 5; i \* i <= num; i += 6) {

**if** (num % i == 0 || num % (i + 2) == 0) {

**return** **false**;

}

}

**return** **true**;

}}

Program:

A screenshot of a computer

Description automatically generated

4. Write a java program to implement integer wrapper class methods. (Any 5 methods)

CODE :

**package** lab6;

**public** **class** IntegerWrapperMethods {

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

// Example values

**int** num1 = 123;

**int** num2 = -456;

// 1. Integer.toString(int i)

String num1AsString = Integer.*toString*(num1);

System.***out***.println("1. Integer.toString(int i) Example:");

System.***out***.println("Integer as String: " + num1AsString);

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

// 2. Integer.parseInt(String s)

String num2AsString = "-789";

**int** num2Parsed = Integer.*parseInt*(num2AsString);

System.***out***.println("2. Integer.parseInt(String s) Example:");

System.***out***.println("String to Integer: " + num2Parsed);

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

// 3. Integer.bitCount(int i)

**int** bitCountNum1 = Integer.*bitCount*(num1);

**int** bitCountNum2 = Integer.*bitCount*(num2);

System.***out***.println("3. Integer.bitCount(int i) Example:");

System.***out***.println("Number of one-bits in " + num1 + ": " + bitCountNum1);

System.***out***.println("Number of one-bits in " + num2 + ": " + bitCountNum2);

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

// 4. Integer.compare(int x, int y)

**int** compareResult1 = Integer.*compare*(num1, num2);

**int** compareResult2 = Integer.*compare*(num2, num1);

**int** compareResult3 = Integer.*compare*(num1, num1);

System.***out***.println("4. Integer.compare(int x, int y) Example:");

System.***out***.println("Comparing " + num1 + " and " + num2 + ": " + compareResult1);

System.***out***.println("Comparing " + num2 + " and " + num1 + ": " + compareResult2);

System.***out***.println("Comparing " + num1 + " and " + num1 + ": " + compareResult3);

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

// 5. Integer.toHexString(int i)

**int** num3 = 255;

String hexString = Integer.*toHexString*(num3);

System.***out***.println("5. Integer.toHexString(int i) Example:");

System.***out***.println("Hexadecimal representation of " + num3 + ": " + hexString);

}

}

Program:

A screenshot of a computer

Description automatically generated

5. Write a java program to implement double wrapper class methods. (Any 5 methods)

CODE :

**package** lab6;

**public** **class** DoubleWrapperMethods {

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

// Example values

**double** num1 = 123.45;

**double** num2 = -67.89;

// 1. Double.toString(double d)

String num1AsString = Double.*toString*(num1);

System.***out***.println("1. Double.toString(double d) Example:");

System.***out***.println("Double as String: " + num1AsString);

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

// 2. Double.parseDouble(String s)

String num2AsString = "-987.65";

**double** num2Parsed = Double.*parseDouble*(num2AsString);

System.***out***.println("2. Double.parseDouble(String s) Example:");

System.***out***.println("String to Double: " + num2Parsed);

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

// 3. Double.isNaN(double d)

**double** num3 = Double.***NaN***;

**boolean** isNaNNum3 = Double.*isNaN*(num3);

System.***out***.println("3. Double.isNaN(double d) Example:");

System.***out***.println("Is " + num3 + " NaN? " + isNaNNum3);

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

// 4. Double.compare(double d1, double d2)

**int** compareResult1 = Double.*compare*(num1, num2);

**int** compareResult2 = Double.*compare*(num2, num1);

**int** compareResult3 = Double.*compare*(num1, num1);

System.***out***.println("4. Double.compare(double d1, double d2) Example:");

System.***out***.println("Comparing " + num1 + " and " + num2 + ": " + compareResult1);

System.***out***.println("Comparing " + num2 + " and " + num1 + ": " + compareResult2);

System.***out***.println("Comparing " + num1 + " and " + num1 + ": " + compareResult3);

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

// 5. Double.isFinite(double d)

**double** num4 = Double.***POSITIVE\_INFINITY***;

**boolean** isFiniteNum4 = Double.*isFinite*(num4);

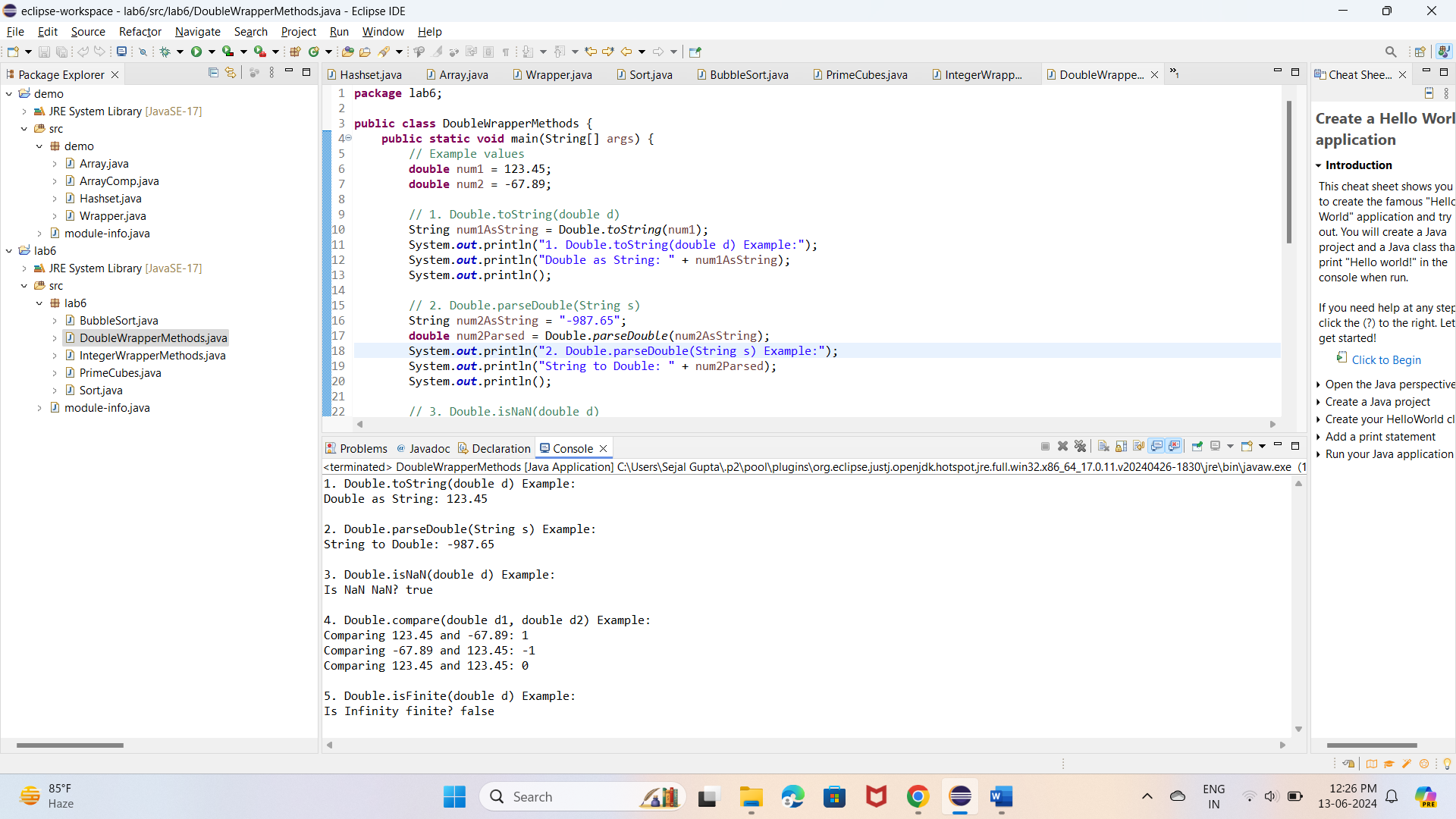
System.***out***.println("5. Double.isFinite(double d) Example:");

System.***out***.println("Is " + num4 + " finite? " + isFiniteNum4);

}

}

Program:



6. Write a java program to implement float wrapper class methods. (Any 5 methods)

CODE :

**package** lab6;

**public** **class** FloatWrapperMethods {

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

// Example values

**float** num1 = 123.45f;

**float** num2 = -67.89f;

// 1. Float.toString(float f)

String num1AsString = Float.*toString*(num1);

System.***out***.println("1. Float.toString(float f) Example:");

System.***out***.println("Float as String: " + num1AsString);

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

// 2. Float.parseFloat(String s)

String num2AsString = "-987.65";

**float** num2Parsed = Float.*parseFloat*(num2AsString);

System.***out***.println("2. Float.parseFloat(String s) Example:");

System.***out***.println("String to Float: " + num2Parsed);

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

// 3. Float.isNaN(float f)

**float** num3 = Float.***NaN***;

**boolean** isNaNNum3 = Float.*isNaN*(num3);

System.***out***.println("3. Float.isNaN(float f) Example:");

System.***out***.println("Is " + num3 + " NaN? " + isNaNNum3);

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

// 4. Float.compare(float f1, float f2)

**int** compareResult1 = Float.*compare*(num1, num2);

**int** compareResult2 = Float.*compare*(num2, num1);

**int** compareResult3 = Float.*compare*(num1, num1);

System.***out***.println("4. Float.compare(float f1, float f2) Example:");

System.***out***.println("Comparing " + num1 + " and " + num2 + ": " + compareResult1);

System.***out***.println("Comparing " + num2 + " and " + num1 + ": " + compareResult2);

System.***out***.println("Comparing " + num1 + " and " + num1 + ": " + compareResult3);

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

// 5. Float.isFinite(float f)

**float** num4 = Float.***POSITIVE\_INFINITY***;

**boolean** isFiniteNum4 = Float.*isFinite*(num4);

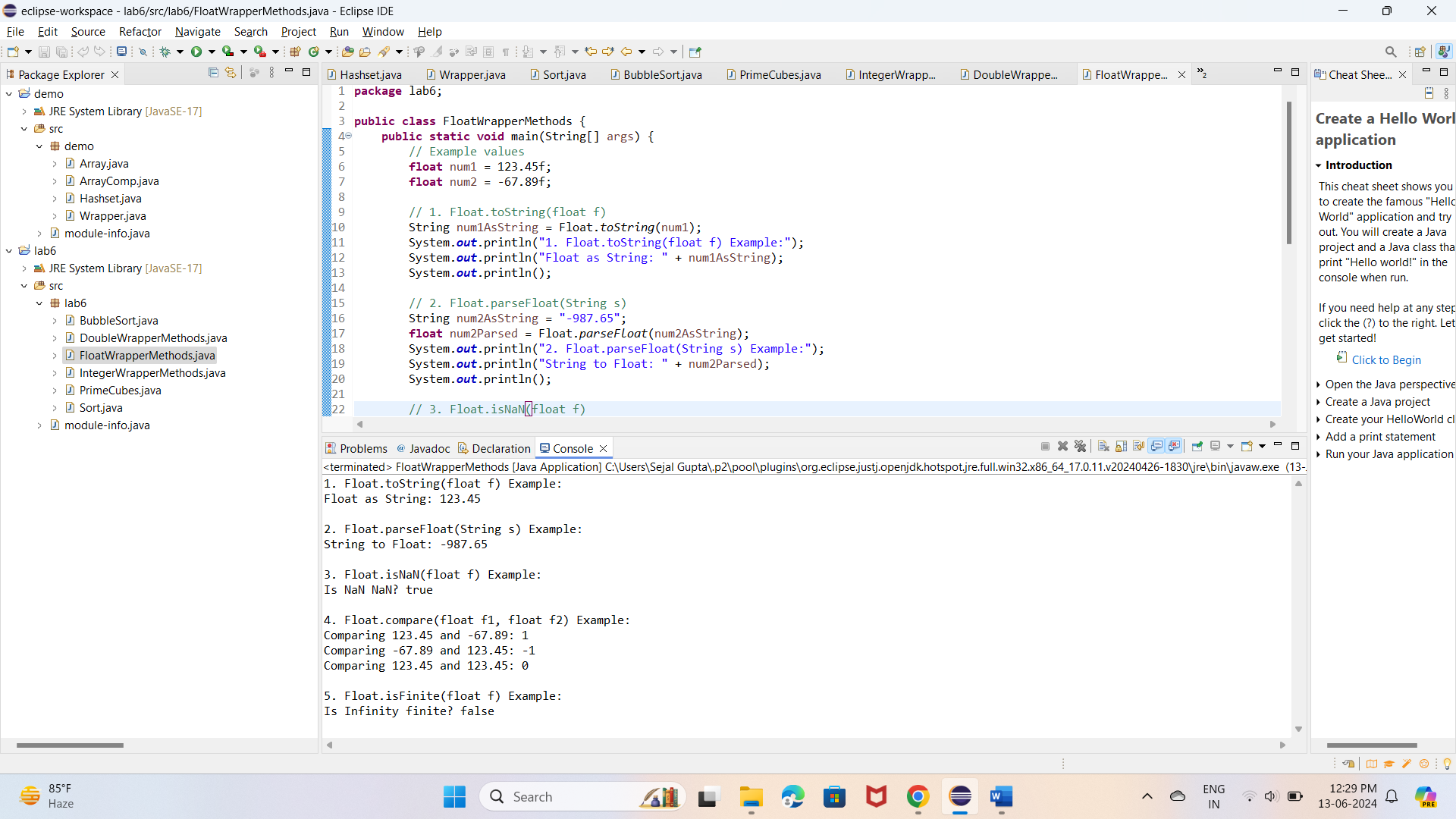
System.***out***.println("5. Float.isFinite(float f) Example:");

System.***out***.println("Is " + num4 + " finite? " + isFiniteNum4);

}

}

Program:



7. Write a Java program to validate email addresses using regular expressions. The email should have the format username@domain.com where username and domain can contain alphanumeric characters, dots, and hyphens.

CODE:

**package** lab6;

**import** java.util.regex.Matcher;

**import** java.util.regex.Pattern;

**public** **class** EmailValidator {

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

// Test cases

String email1 = "user123@example.com";

String email2 = "user.name@domain.co.in";

String email3 = "user-name@sub-domain.domain.com";

String email4 = "user123@example"; // Invalid email format

String email5 = "user!name@example.com"; // Invalid email format

System.***out***.println(email1 + " is valid? " + *isValidEmail*(email1));

System.***out***.println(email2 + " is valid? " + *isValidEmail*(email2));

System.***out***.println(email3 + " is valid? " + *isValidEmail*(email3));

System.***out***.println(email4 + " is valid? " + *isValidEmail*(email4));

System.***out***.println(email5 + " is valid? " + *isValidEmail*(email5));

}

// Method to validate an email address

**public** **static** **boolean** isValidEmail(String email) {

// Regular expression for email validation

String regex = "^[a-zA-Z0-9.\_-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,}$";

// Compile the regex pattern

Pattern pattern = Pattern.*compile*(regex);

// Create matcher object to match given email with regex

Matcher matcher = pattern.matcher(email);

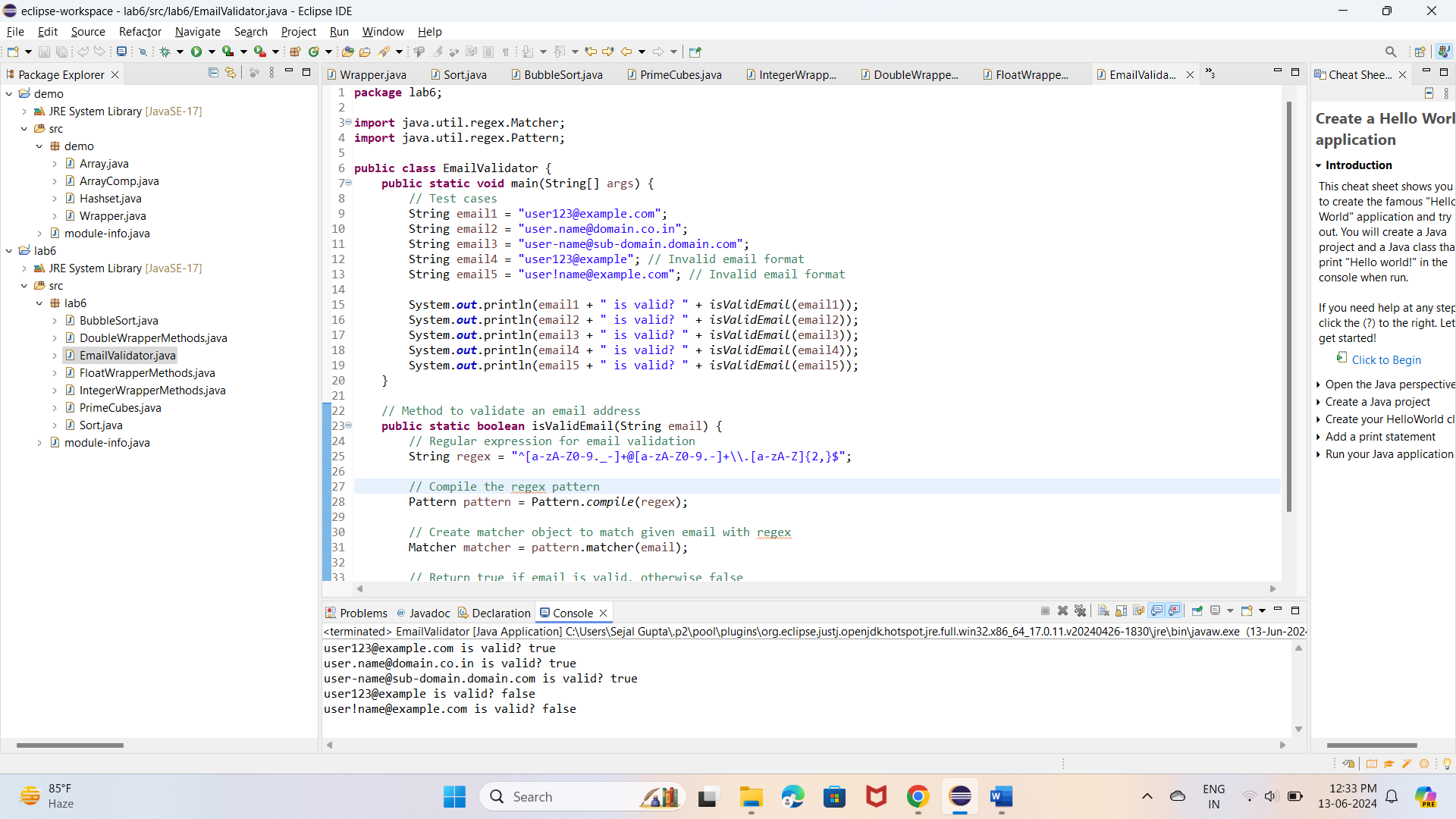
// Return true if email is valid, otherwise false

**return** matcher.matches();

}

}

Program:



8. Create a Java program to validate phone numbers. The format should be (xxx) xxx-xxxx where x is a digit.

CODE :

**package** lab6;

**import** java.util.Scanner;

**import** java.util.regex.Matcher;

**import** java.util.regex.Pattern;

**public** **class** PhoneNumberValidator {

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

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

System.***out***.print("Enter a phone number (in the format (xxx) xxx-xxxx): ");

String phoneNumber = scanner.nextLine();

scanner.close();

**if** (*isValidPhoneNumber*(phoneNumber)) {

System.***out***.println("Valid phone number: " + phoneNumber);

} **else** {

System.***out***.println("Invalid phone number: " + phoneNumber);

}

}

// Method to validate a phone number

**public** **static** **boolean** isValidPhoneNumber(String phoneNumber) {

// Regular expression for phone number validation

String regex = "^\\(\\d{3}\\) \\d{3}-\\d{4}$";

// Compile the regex pattern

Pattern pattern = Pattern.*compile*(regex);

// Create matcher object to match given phone number with regex

Matcher matcher = pattern.matcher(phoneNumber);

// Return true if phone number is valid, otherwise false

**return** matcher.matches();

}

}

Program:

