**LAB-6**

1. Write a java 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.

Program-

**package** lab6;

**import** java.util.Arrays;

**public** **class** SortArrays {

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

// **TODO** Auto-generated method stub

// Define arrays

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

String[] sArray = {"C", "O", "I", "P", "U"};

// Sort arrays

Arrays.*sort*(iArray);

Arrays.*sort*(sArray);

// Print sorted arrays

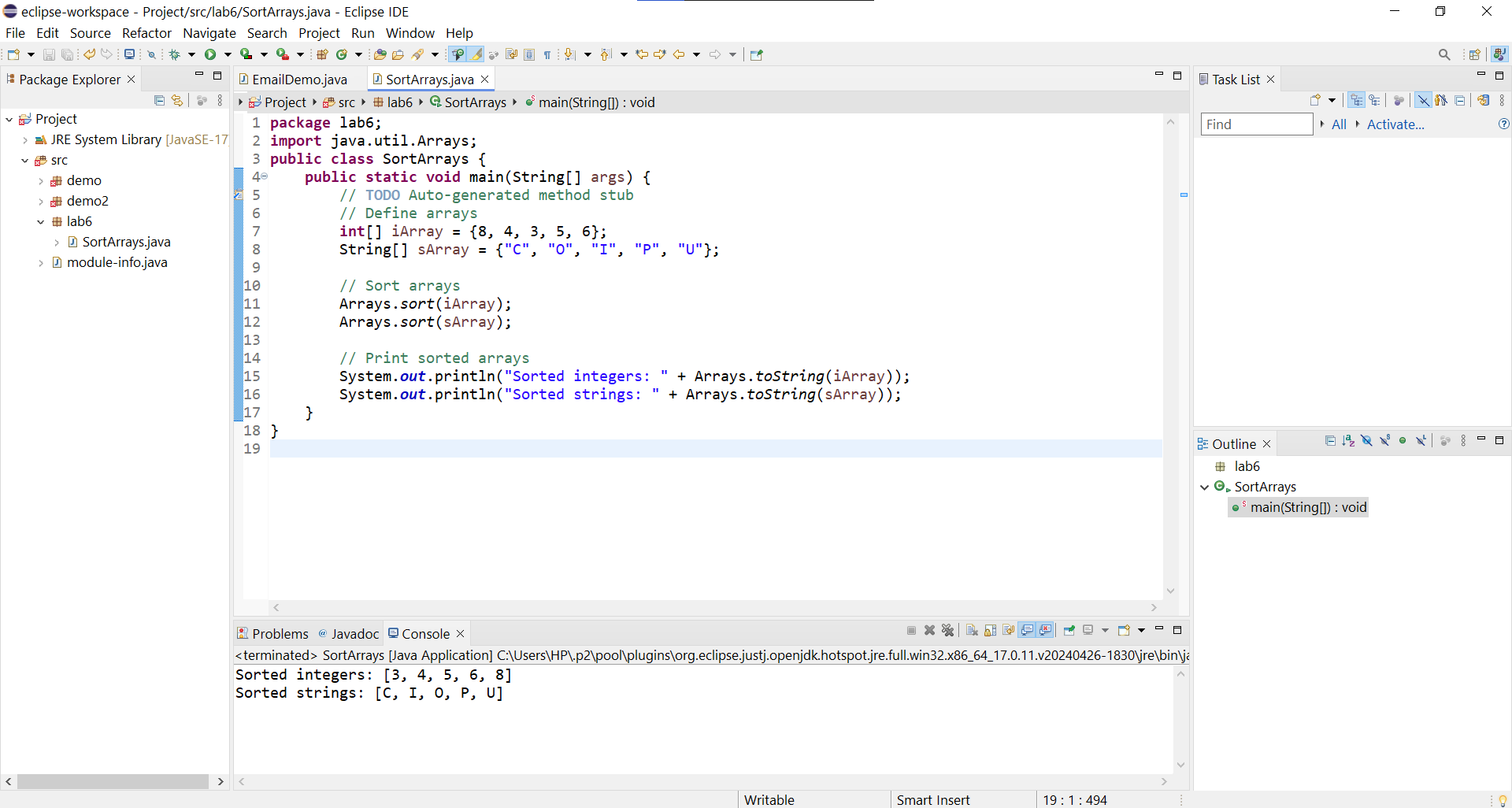
System.***out***.println("Sorted integers: " + Arrays.*toString*(iArray));

System.***out***.println("Sorted strings: " + Arrays.*toString*(sArray));

}

}

Output-



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

Program-

**package** lab6;

**import** java.util.Scanner;

**import** java.util.Arrays;

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

//Method to perform bubble sort

**public** **static** **void** bubbleSort(**int**[] array)

{

**int** n=array.length;

**boolean** swap;

//outer loop to traverse through all array elements

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

swap=**false**;

//Inner loop to compare adjacent elements

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

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

//swap if the element found is greater than the next element

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

array[j]=array[j+1];

array[j+1]=temp;

swap=**true**;

}

}

//if no two elements are swapped by inner loop, then break

**if**(!swap) **break**;

}

}

//method to print the array

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

**for**(**int** i:array) {

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

}

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

}

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

// **TODO** Auto-generated method stub

**int**[] array = {56, 23, 62, 12, 20, 98, 4};

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

*printArray*(array);

*bubbleSort*(array);

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

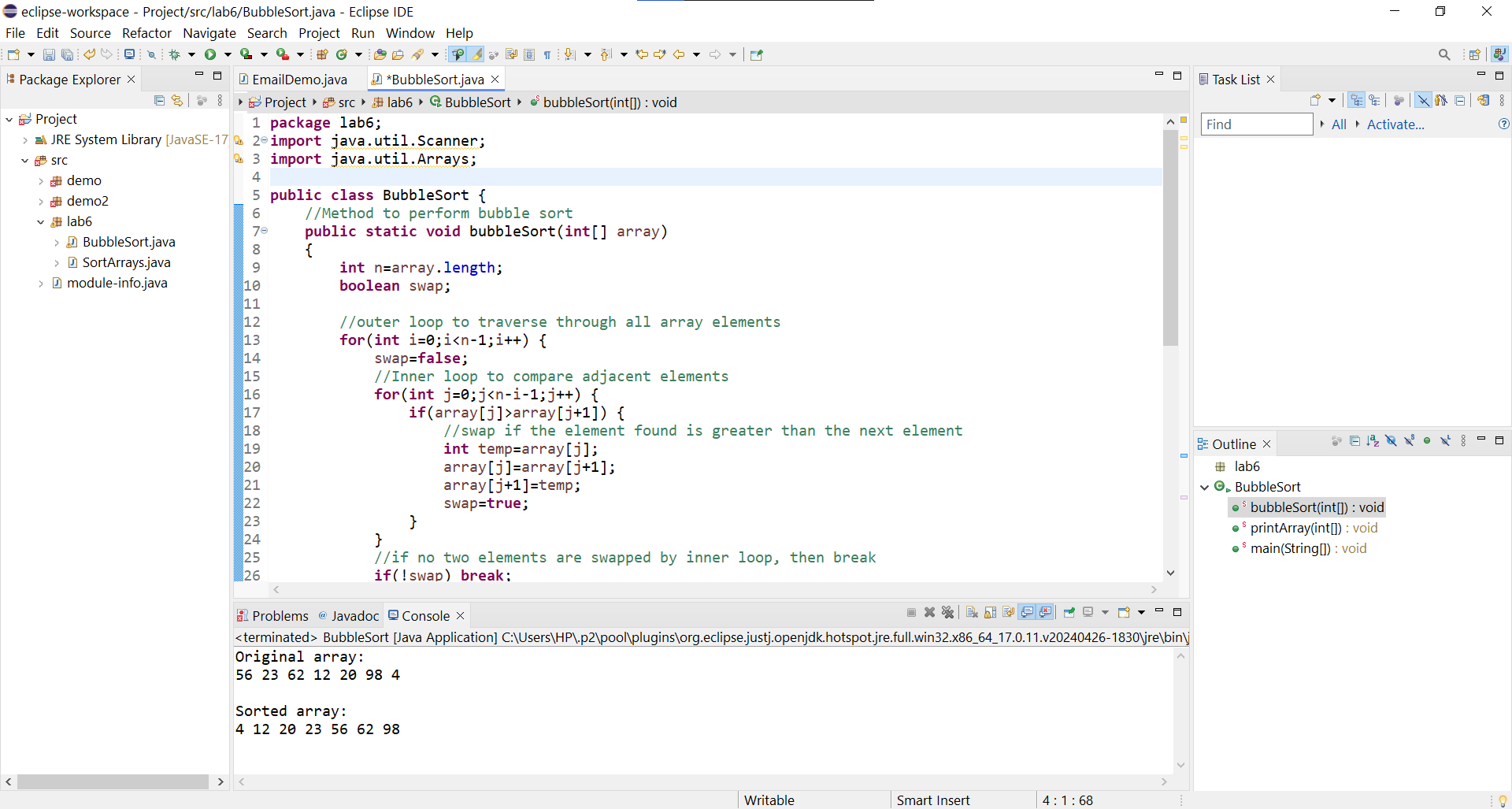
System.***out***.println("Sorted array:");

*printArray*(array);

}

}

Output-



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

Program-

**package** lab6;

**import** java.util.Scanner;

**public** **class** Cube {

// Method to check if a number is prime

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

**if** (num <= 1) **return** **false**;

**for** (**int** i = 2; i <= Math.*sqrt*(num); i++) {

**if** (num % i == 0) **return** **false**;

}

**return** **true**;

}

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

// **TODO** Auto-generated method stub

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

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

// Input 10 elements from the user

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

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

n[i] = scanner.nextInt();

}

// Process the array and print the cube of prime numbers

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

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

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

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

System.***out***.println("Prime No: " + num + ", Cube: " + cube);

}

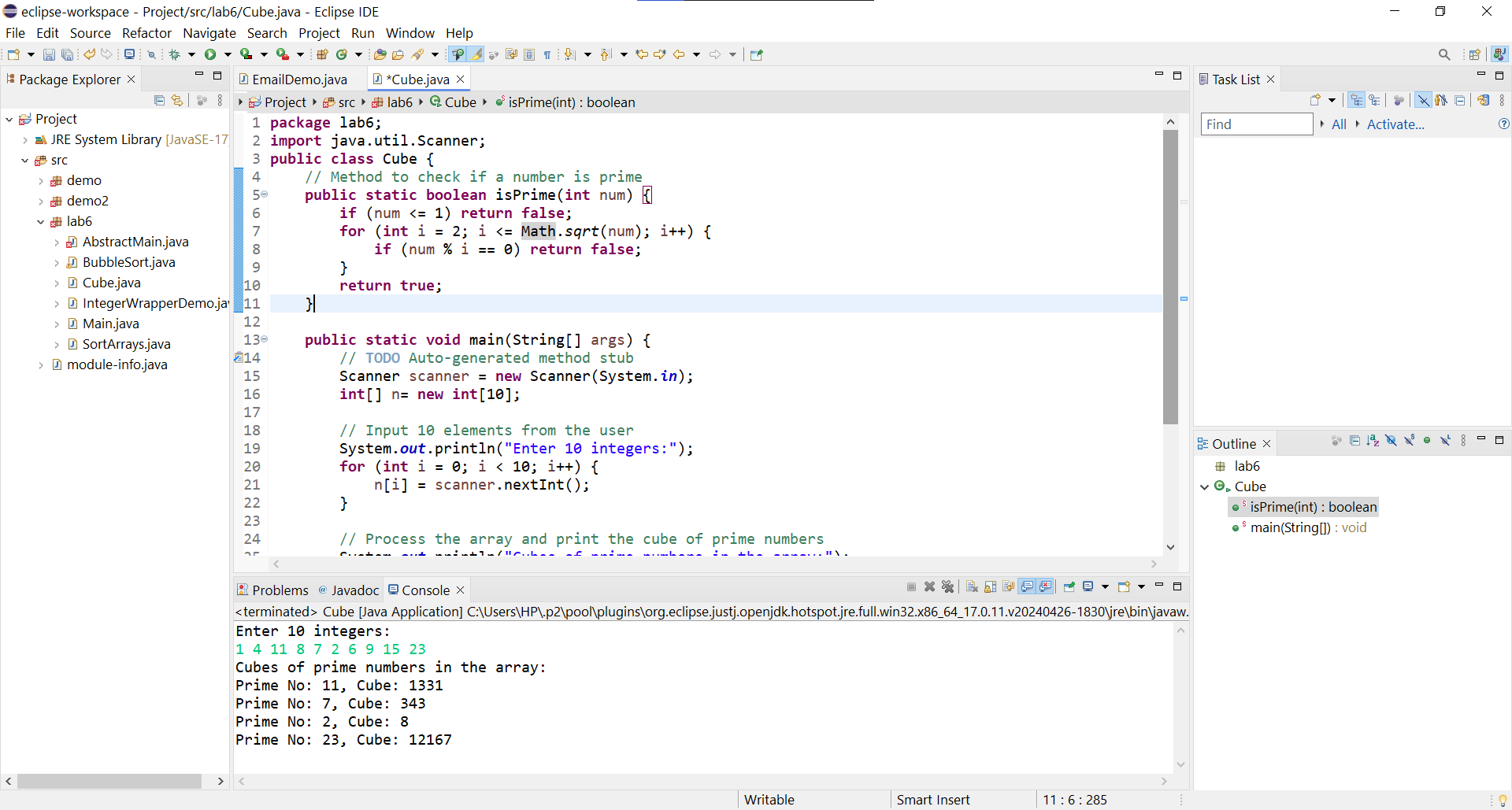
}

scanner.close();

}

}

Output-



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

Program-

**package** lab6;

**public** **class** IntegerWrapperDemo {

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

// **TODO** Auto-generated method stub

// parseInt(String s)

String str = "23";

**int** parsedInt = Integer.*parseInt*(str);

System.***out***.println("Parsed integer from string \"" + str + "\": " + parsedInt);

// valueOf(String s)

Integer integerValue = Integer.*valueOf*("456");

System.***out***.println("Integer object from string \"456\": " + integerValue);

// toString(int i)

**int** number = 789;

String numberString = Integer.*toString*(number);

System.***out***.println("String representation of integer " + number + ": " + numberString);

// compare(int x, int y)

**int** x = 10;

**int** y = 20;

**int** comparisonResult = Integer.*compare*(x, y);

System.***out***.println("Comparison of " + x + " and " + y + ": " + comparisonResult);

// Comparison result is -1 if x < y, 0 if x == y, and 1 if x > y

// intValue()

Integer intObj = 999;

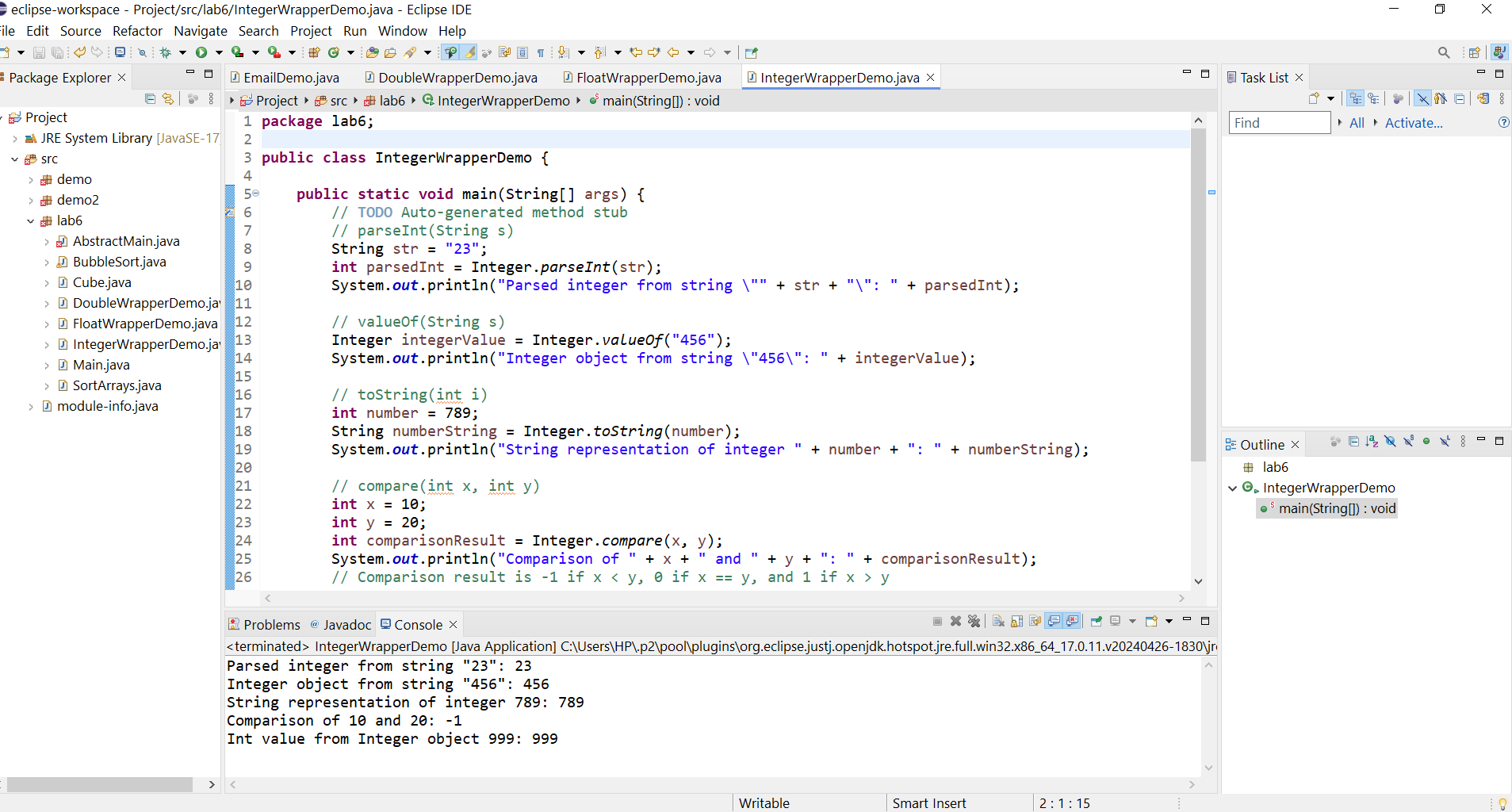
**int** intValue = intObj.intValue();

System.***out***.println("Int value from Integer object " + intObj + ": " + intValue);

}

}

Output-



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

Program-

**package** lab6;

**public** **class** DoubleWrapperDemo {

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

// **TODO** Auto-generated method stub

// parseDouble(String s)

String doubleString = "23.21";

**double** parsedDouble = Double.*parseDouble*(doubleString);

System.***out***.println("Parsed double from string \"" + doubleString + "\": " + parsedDouble);

// valueOf(String s)

Double doubleValue = Double.*valueOf*("456.78");

System.***out***.println("Double object from string \"456.78\": " + doubleValue);

// toString(double d)

**double** number = 789.1;

String numberString = Double.*toString*(number);

System.***out***.println("String representation of double " + number + ": " + numberString);

// compare(double d1, double d2)

**double** d1 = 10.5;

**double** d2 = 30.5;

**int** comparisonResult = Double.*compare*(d1, d2);

System.***out***.println("Comparison of " + d1 + " and " + d2 + ": " + comparisonResult);

// Comparison result is -1 if d1 < d2, 0 if d1 == d2, and 1 if d1 > d2

// isNaN(double v)

**double** nanValue = Double.***NaN***;

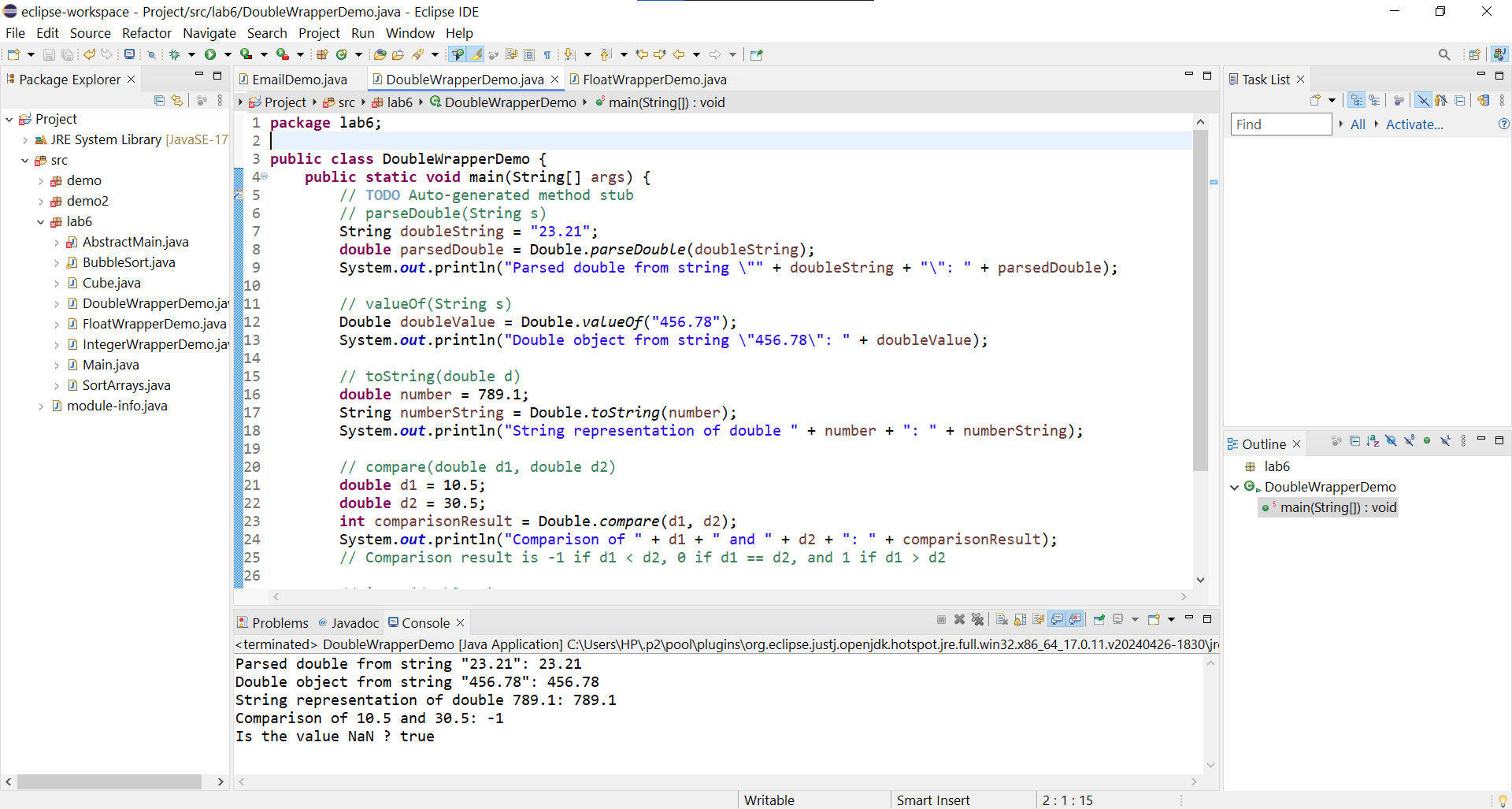
**boolean** isNan = Double.*isNaN*(nanValue);

System.***out***.println("Is the value NaN ? " + isNan);

}

}

Output-



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

Program-

**package** lab6;

**public** **class** FloatWrapperDemo {

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

// **TODO** Auto-generated method stub

// parseFloat(String s)

String floatString = "72.65";

**float** parsedFloat = Float.*parseFloat*(floatString);

System.***out***.println("Parsed float from string \"" + floatString + "\": " + parsedFloat);

// valueOf(String s)

Float floatValue = Float.*valueOf*("456.78");

System.***out***.println("Float object from string \"456.78\": " + floatValue);

// toString(float f)

**float** number = 231.341f;

String numberString = Float.*toString*(number);

System.***out***.println("String representation of float " + number + ": " + numberString);

// compare(float f1, float f2)

**float** f1 = 10.5f;

**float** f2 = 20.5f;

**int** comparisonResult = Float.*compare*(f1, f2);

System.***out***.println("Comparison of " + f1 + " and " + f2 + ": " + comparisonResult);

// Comparison result is -1 if f1 < f2, 0 if f1 == f2, and 1 if f1 > f2

// isNaN(float v)

**float** nanValue = Float.***NaN***;

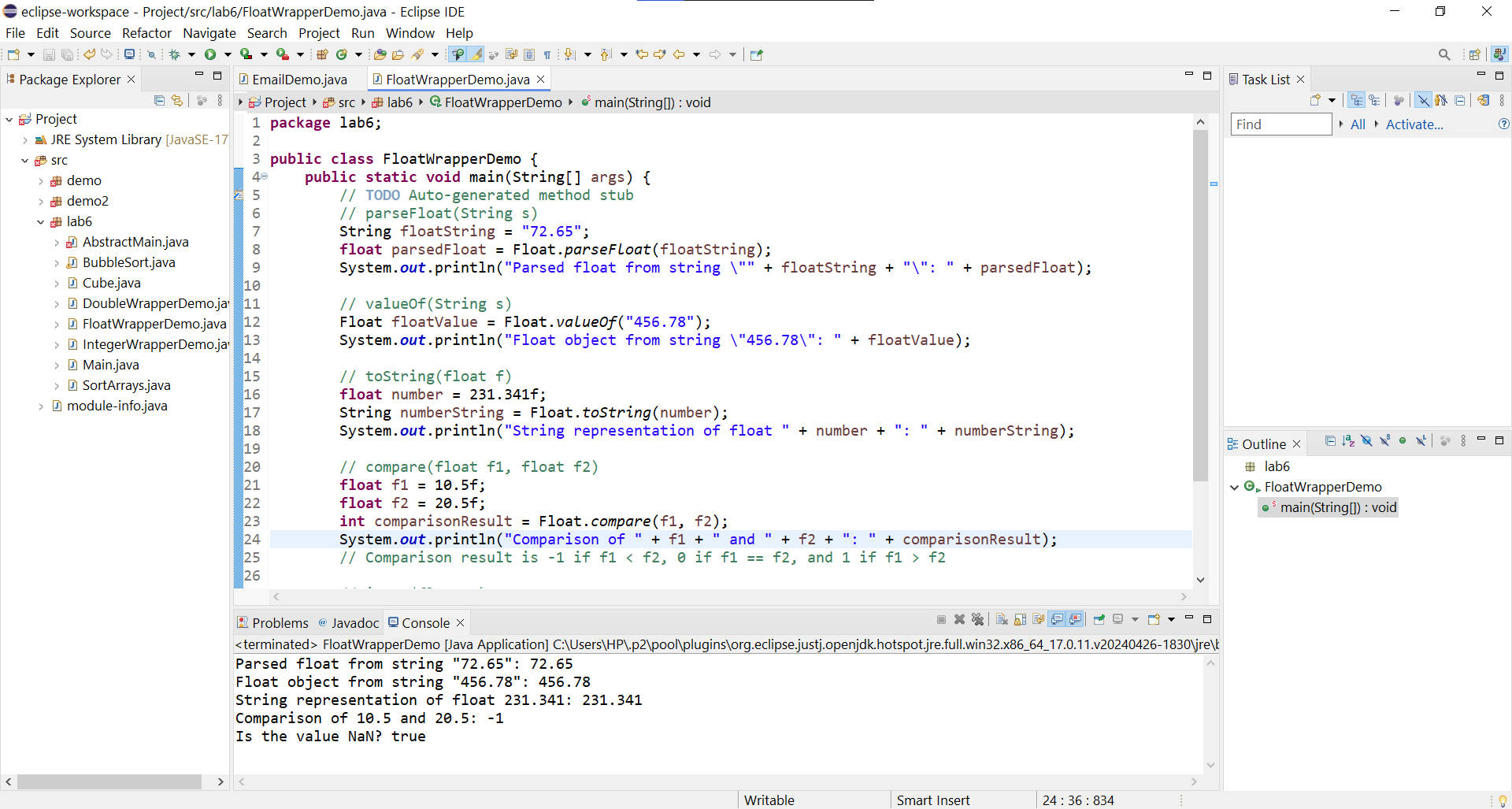
**boolean** isNan = Float.*isNaN*(nanValue);

System.***out***.println("Is the value NaN? " + isNan);

}

}

Output-



1. 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.

Program-

**package** demo2;

**import** java.util.regex.\*;

**public** **class** EmailDemo {

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

// **TODO** Auto-generated method stub

//Test cases

String email="user123@domain.com";

String email2="user!21@demo.com";//Invalid Email

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

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

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

}

//Method to validate Email Address

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

//regular expressions for email validation

String regex="^[\\w.-]+@[\\w.-]+\\.[a-z]{2,}$";

//compile the regex pattern

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

//create matcher object to match email

Matcher matcher=p.matcher(email);

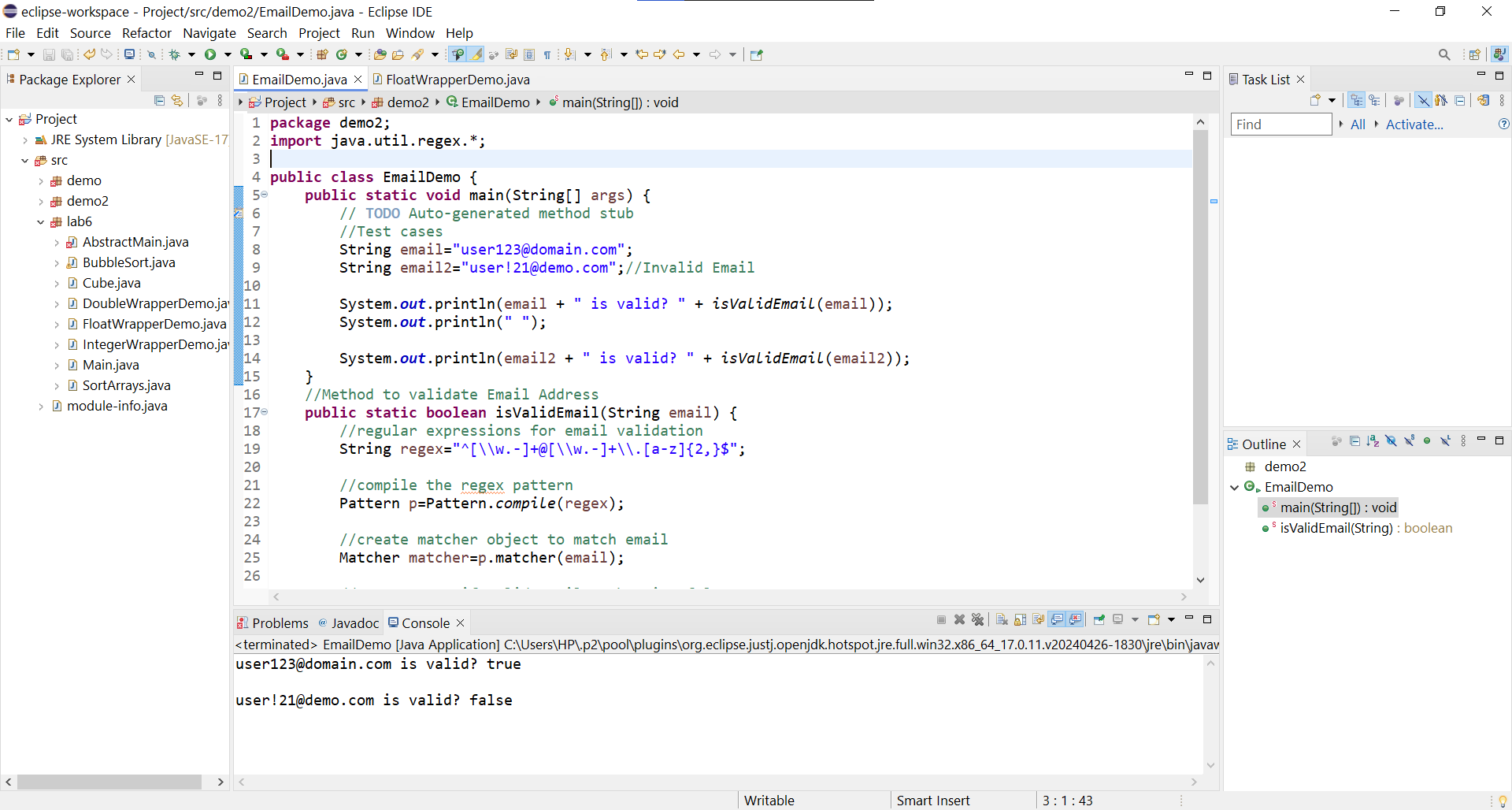
//return true if valid email, otherwise false

**return** matcher.matches();

}

}

Output-



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

Program-

**package** lab6;

**import** java.util.Scanner;

**import** java.util.regex.\*;

**public** **class** PhoneNoValidator {

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

// **TODO** Auto-generated method stub

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

System.***out***.println("Enter a Phone Number in format (xxx) xxx-xxxx : ");

String no=sc.nextLine();

sc.close();

**if**(*isValidPhoneNo*(no)) {

System.***out***.println(no+ " is a Valid Phone Number");

}

**else**

{

System.***out***.println(no+ " is not a Valid Phone Number");

}

}

//method to validate phone number

**public** **static** **boolean** isValidPhoneNo(String no) {

//regular expression for phone number

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

// Compile the regex pattern

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

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

Matcher matcher = p.matcher(no);

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

**return** matcher.matches();

}

}

Output-

