**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.**

**CODE**:-

package demo2;

import java.util.Arrays;

public class SortIntegersAndStrings {

public static void main(String[] args) {

// Sort integers

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

Arrays.*sort*(numbers); // Efficient built-in sorting

System.***out***.print("Sorted integers: ");

for (int num : numbers) {

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

}

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

// Sort strings (case-insensitive)

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

Arrays.*sort*(letters, String.***CASE\_INSENSITIVE\_ORDER***); // Case-insensitive sorting

System.***out***.print("Sorted strings (case-insensitive): ");

for (String letter : letters) {

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

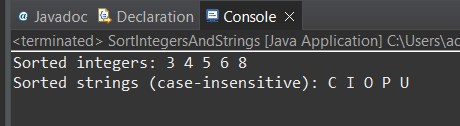
}

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

}

}

**OUTPUT:-**



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

**CODE:-**

package demo2;

public class BubbleSort {

public static void main(String[] args) {

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

*bubbleSort*(numbers);

System.***out***.print("Sorted integers using bubble sort: ");

for (int num : numbers) {

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

}

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

}

public static void bubbleSort(int[] arr) {

int n = arr.length;

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

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

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

int temp = arr[j];

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

arr[j + 1] = temp;

}

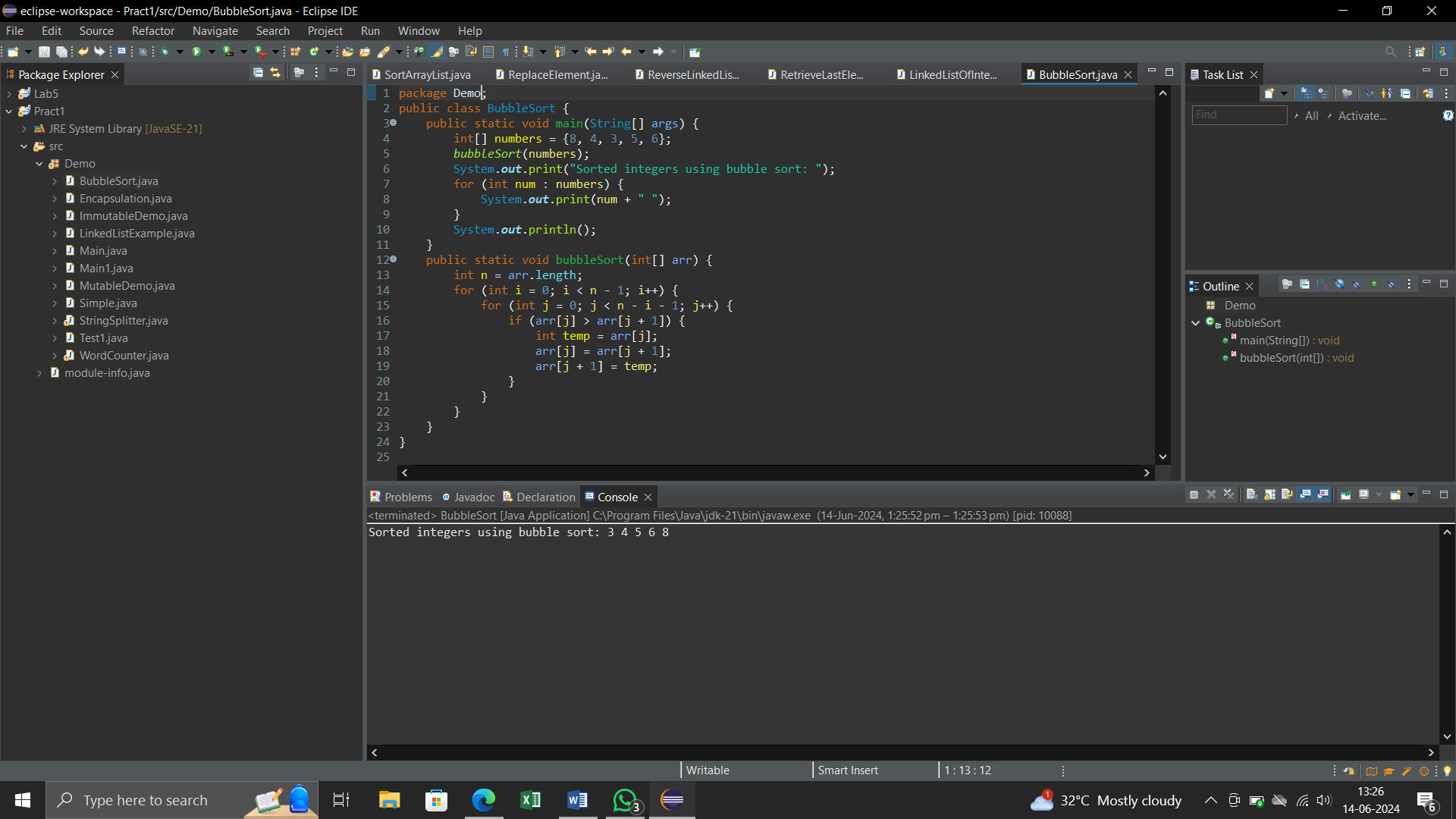
}

}

}

}

**OUTPUT:-**

**

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

**CODE:-**

package demo2;

import java.util.Scanner;

public class PrimeCube {

public static void main(String[] args) {

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

int[] numbers = new int[10];

// Input array of 10 elements

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

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

numbers[i] = scanner.nextInt();

}

// Process and print cubes of prime numbers

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

for (int number : numbers) {

if (*isPrime*(number)) {

System.***out***.println("Number: " + number + ", Cube: " + (number \* number \* number));

}

}

scanner.close();

}

// Function 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;

}

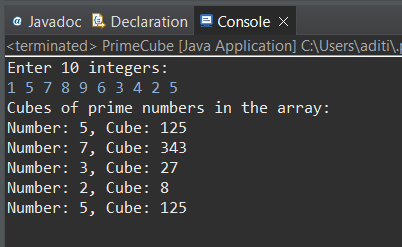
}

return true;

}

}

**OUTPUT:-**



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

**CODE:-**

package demo;

public class IntegerWrapperMethods {

public static void main(String[] args) {

// Creating Integer objects

Integer num1 = 10;

Integer num2 = 20;

// toString() method

System.***out***.println("num1 as string: " + num1.toString());

// intValue() method

int value = num1.intValue();

System.***out***.println("num1 as int value: " + value);

// compareTo() method

int compareResult = num1.compareTo(num2);

if (compareResult == 0) {

System.***out***.println("num1 is equal to num2");

} else if (compareResult < 0) {

System.***out***.println("num1 is less than num2");

} else {

System.***out***.println("num1 is greater than num2");

}

// equals() method

boolean isEqual = num1.equals(num2);

System.***out***.println("Are num1 and num2 equal? " + isEqual);

// valueOf() method

Integer num3 = Integer.*valueOf*("30");

System.***out***.println("num3: " + num3);

// parseInt() method

int parsedInt = Integer.*parseInt*("40");

System.***out***.println("parsedInt: " + parsedInt);

// MAX\_VALUE and MIN\_VALUE constants

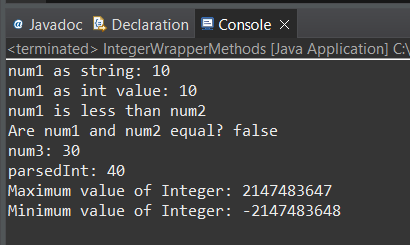
System.***out***.println("Maximum value of Integer: " + Integer.***MAX\_VALUE***);

System.***out***.println("Minimum value of Integer: " + Integer.***MIN\_VALUE***);

}

}

**OUTPUT:-**



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

**CODE:-**

package demo;

public class DoubleWrapperMethods {

public static void main(String[] args) {

// Creating Double objects

Double num1 = 10.5;

Double num2 = 20.7;

// toString() method

System.***out***.println("num1 as string: " + num1.toString());

// doubleValue() method

double value = num1.doubleValue();

System.***out***.println("num1 as double value: " + value);

// compareTo() method

int compareResult = num1.compareTo(num2);

if (compareResult == 0) {

System.***out***.println("num1 is equal to num2");

} else if (compareResult < 0) {

System.***out***.println("num1 is less than num2");

} else {

System.***out***.println("num1 is greater than num2");

}

// equals() method

boolean isEqual = num1.equals(num2);

System.***out***.println("Are num1 and num2 equal? " + isEqual);

// valueOf() method

Double num3 = Double.*valueOf*("30.2");

System.***out***.println("num3: " + num3);

// parseDouble() method

double parsedDouble = Double.*parseDouble*("40.5");

System.***out***.println("parsedDouble: " + parsedDouble);

// MAX\_VALUE and MIN\_VALUE constants

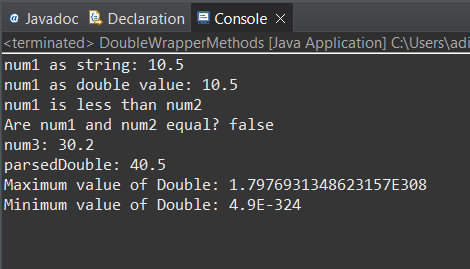
System.***out***.println("Maximum value of Double: " + Double.***MAX\_VALUE***);

System.***out***.println("Minimum value of Double: " + Double.***MIN\_VALUE***);

}

}

**OUTPUT:-**



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

**CODE:-**

package demo;

public class FloatWrapperDemo {

public static void main(String[] args) {

// Creating Float objects

Float floatObj1 = new Float(3.14f);

Float floatObj2 = new Float("5.67");

// Displaying the values

System.***out***.println("floatObj1: " + floatObj1);

System.***out***.println("floatObj2: " + floatObj2);

// Demonstrating some methods

System.***out***.println("floatObj1.compareTo(floatObj2): " + floatObj1.compareTo(floatObj2));

System.***out***.println("Float.compare(floatObj1, floatObj2): " + Float.*compare*(floatObj1, floatObj2));

System.***out***.println("floatObj1.equals(floatObj2): " + floatObj1.equals(floatObj2));

System.***out***.println("floatObj1.floatValue(): " + floatObj1.floatValue());

System.***out***.println("floatObj1.intValue(): " + floatObj1.intValue());

System.***out***.println("floatObj1.toString(): " + floatObj1.toString());

System.***out***.println("Float.toHexString(floatObj1): " + Float.*toHexString*(floatObj1));

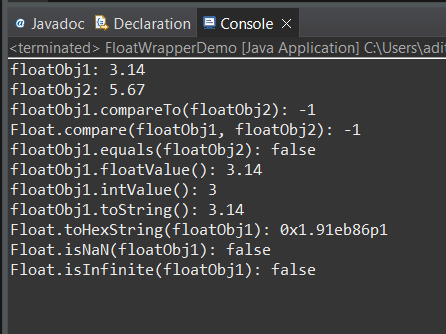
System.***out***.println("Float.isNaN(floatObj1): " + Float.*isNaN*(floatObj1));

System.***out***.println("Float.isInfinite(floatObj1): " + Float.*isInfinite*(floatObj1));

}

}

**OUTPUT:-**



**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 demo2;

import java.util.Scanner;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class EmailValidator {

public static void main(String[] args) {

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

System.***out***.println("Enter an email address to validate:");

String email = scanner.nextLine();

// Define the regex pattern for email validation

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

// Compile the regex pattern

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

// Match the input email with the pattern

Matcher matcher = pattern.matcher(email);

// Validate the email and print the result

if (matcher.matches()) {

System.***out***.println(email + " is a valid email address.");

} else {

System.***out***.println(email + " is not a valid email address.");

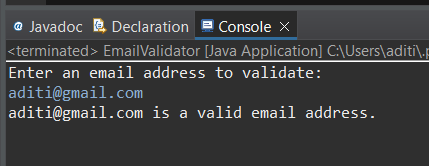
}

scanner.close();

}

}

**OUTPUT:-**



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

**CODE:-**

package demo2;

import java.util.Scanner;

import java.util.regex.Pattern;

import java.util.regex.Matcher;

public class PhoneNumber {

// Regular expression to match the phone number format (xxx) xxx-xxxx

private static final String ***PHONE\_NUMBER\_REGEX*** = "\\+91[- ]?\\d{10}";

public static void main(String[] args) {

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

System.***out***.println("Enter a phone number to validate (format: (xx) xxxxxxxxxx):");

String phoneNumber = scanner.nextLine();

if (*isValidPhoneNumber*(phoneNumber)) {

System.***out***.println("The phone number is valid.");

} else {

System.***out***.println("The phone number is invalid.");

}

scanner.close();

}

// Method to validate the phone number using regular expression

public static boolean isValidPhoneNumber(String phoneNumber) {

Pattern pattern = Pattern.*compile*(***PHONE\_NUMBER\_REGEX***);

Matcher matcher = pattern.matcher(phoneNumber);

return matcher.matches();

}

}

**OUTPUT:-**

