**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.s**

**Code:**

import java.util.Arrays;

public class SortExample {

public static void main(String[] args) {

// Initialize the arrays

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

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

// Sort the arrays

Arrays.sort(intArray);

Arrays.sort(strArray);

// Print the sorted arrays

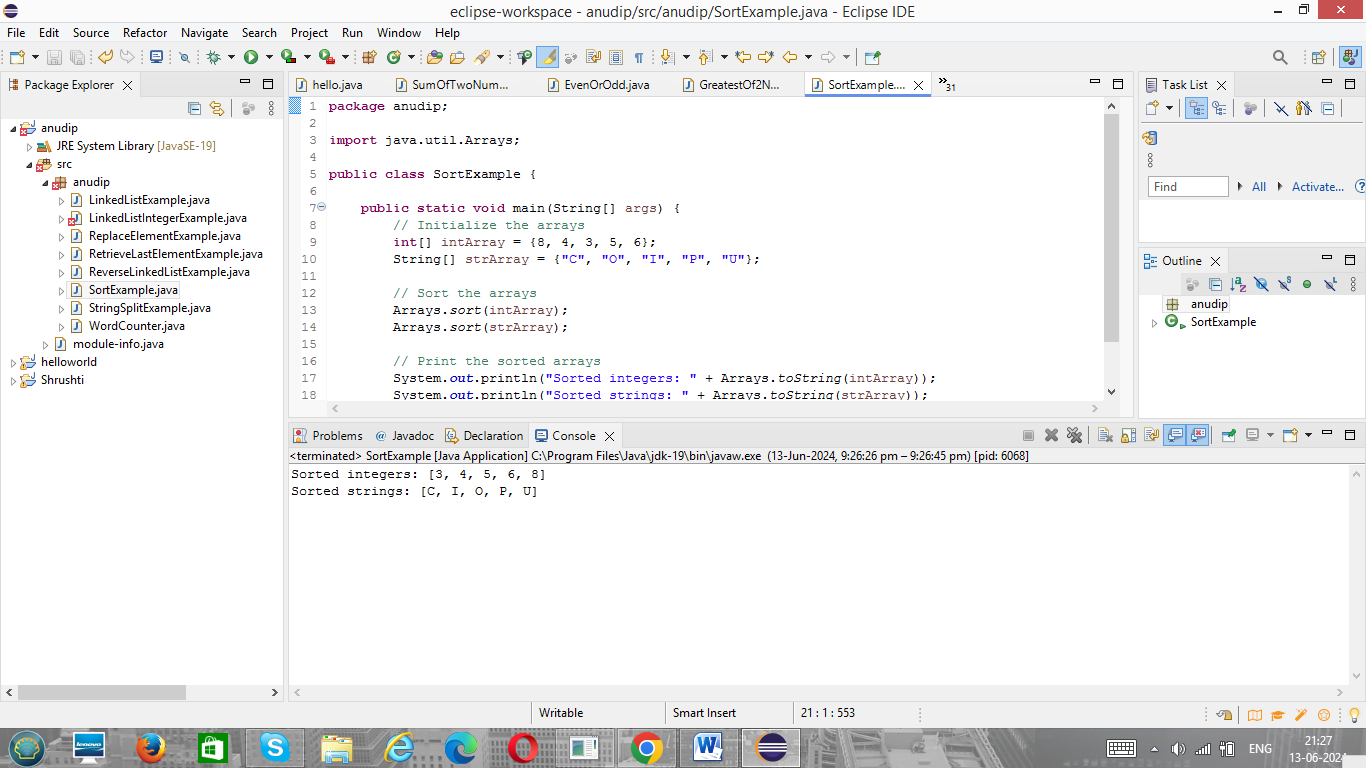
System.out.println("Sorted integers: " + Arrays.toString(intArray));

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

}

}

**Output:-**



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

**Code:**

public class BubbleSortExample {

public static void main(String[] args) {

// Initialize the array

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

// Call the bubble sort method

bubbleSort(intArray);

// Print the sorted array

System.out.println("Sorted integers: ");

for (int num : intArray) {

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

}

}

// Method to implement bubble sort

public static void bubbleSort(int[] array) {

int n = array.length;

boolean swapped;

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

swapped = false;

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

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

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

int temp = array[j];

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

array[j + 1] = temp;

swapped = true;

}

}

// If no elements were swapped, the array is already sorted

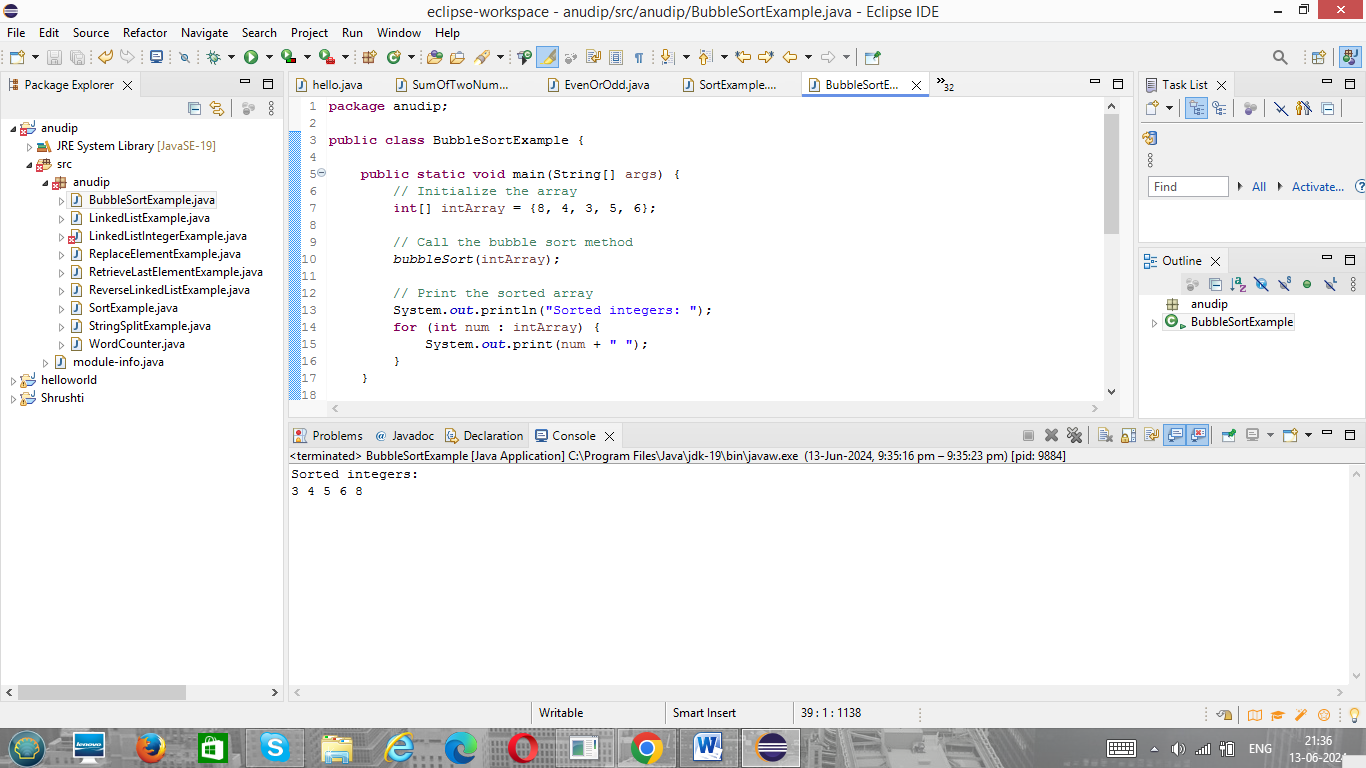
if (!swapped) break;

}

}

}

**Output:**



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

**Code:**

import java.util.Scanner;

public class PrimeNumberCubes {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int[] array = new int[10];

// Input 10 elements into the array

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

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

array[i] = scanner.nextInt();

}

// Process and print the cube of prime numbers

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

for (int num : array) {

if (isPrime(num)) {

System.out.println(num + "^3 = " + cube(num));

}

}

}

// 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;

}

// Method to calculate the cube of a number

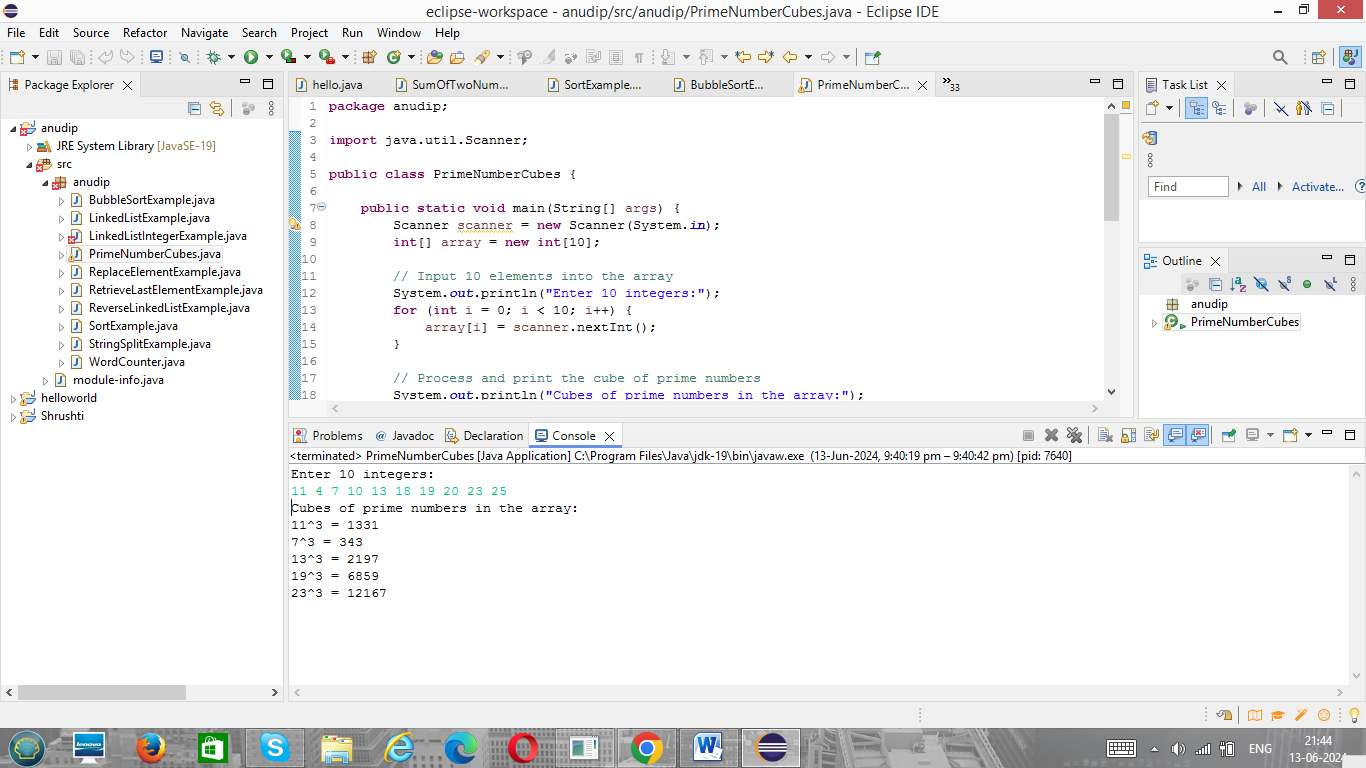
public static int cube(int num) {

return num \* num \* num;

}

}

**Output:**



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

**Code:-**

public class IntegerWrapperExample {

public static void main(String[] args) {

// Demonstrating parseInt(String s)

String numberStr = "123";

int number = Integer.parseInt(numberStr);

System.out.println("Using parseInt: The integer value is " + number);

// Demonstrating valueOf(String s)

Integer numberObj = Integer.valueOf(numberStr);

System.out.println("Using valueOf: The Integer object is " + numberObj);

// Demonstrating toString()

String numberObjStr = numberObj.toString();

System.out.println("Using toString: The String representation is " + numberObjStr);

// Demonstrating compareTo(Integer anotherInteger)

Integer anotherNumberObj = 150;

int comparisonResult = numberObj.compareTo(anotherNumberObj);

if (comparisonResult < 0) {

System.out.println("Using compareTo: " + numberObj + " is less than " + anotherNumberObj);

} else if (comparisonResult > 0) {

System.out.println("Using compareTo: " + numberObj + " is greater than " + anotherNumberObj);

} else {

System.out.println("Using compareTo: " + numberObj + " is equal to " + anotherNumberObj);

}

// Demonstrating intValue()

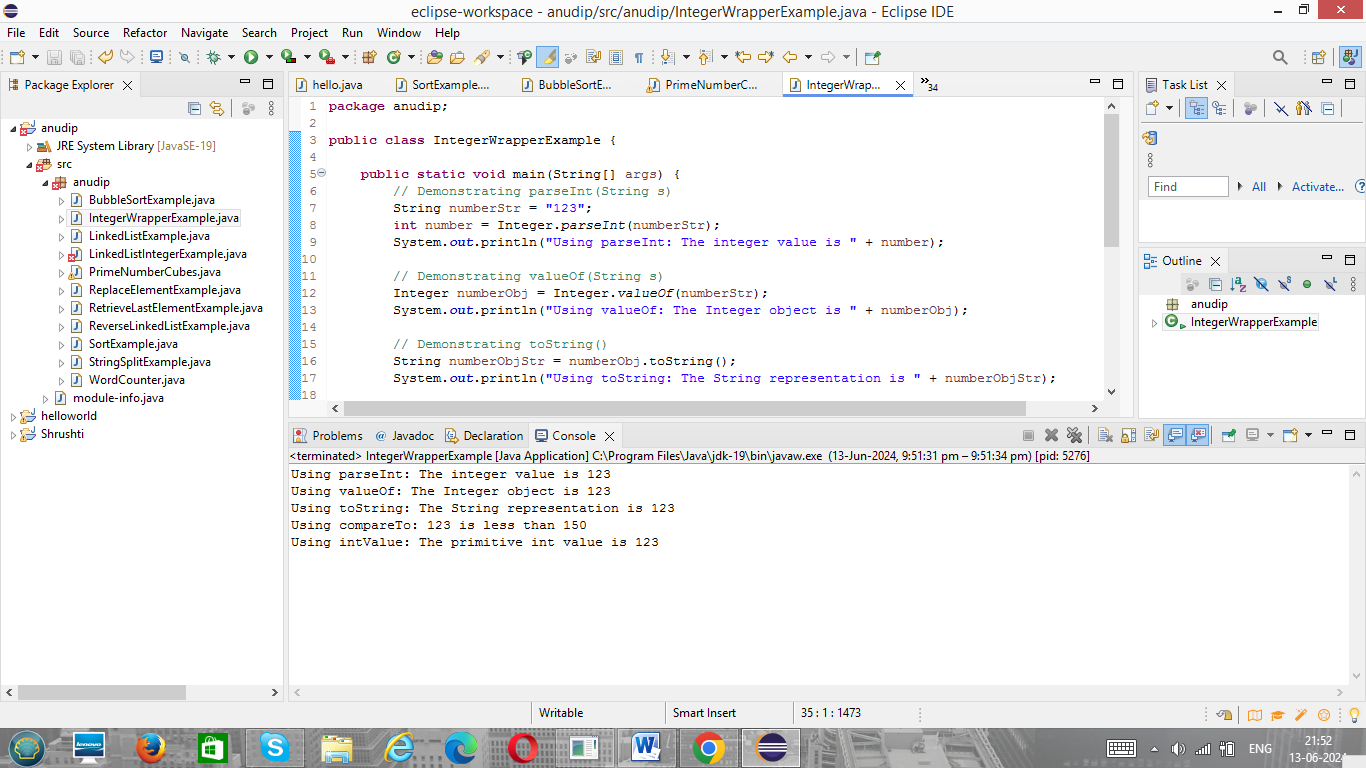
int primitiveInt = numberObj.intValue();

System.out.println("Using intValue: The primitive int value is " + primitiveInt);

}

}

**Output:**



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

**Code:**

public class DoubleWrapperExample {

public static void main(String[] args) {

// Demonstrating parseDouble(String s)

String doubleStr = "123.45";

double number = Double.parseDouble(doubleStr);

System.out.println("Using parseDouble: The double value is " + number);

// Demonstrating valueOf(String s)

Double numberObj = Double.valueOf(doubleStr);

System.out.println("Using valueOf: The Double object is " + numberObj);

// Demonstrating toString()

String numberObjStr = numberObj.toString();

System.out.println("Using toString: The String representation is " + numberObjStr);

// Demonstrating compareTo(Double anotherDouble)

Double anotherNumberObj = 150.75;

int comparisonResult = numberObj.compareTo(anotherNumberObj);

if (comparisonResult < 0) {

System.out.println("Using compareTo: " + numberObj + " is less than " + anotherNumberObj);

} else if (comparisonResult > 0) {

System.out.println("Using compareTo: " + numberObj + " is greater than " + anotherNumberObj);

} else {

System.out.println("Using compareTo: " + numberObj + " is equal to " + anotherNumberObj);

}

// Demonstrating doubleValue()

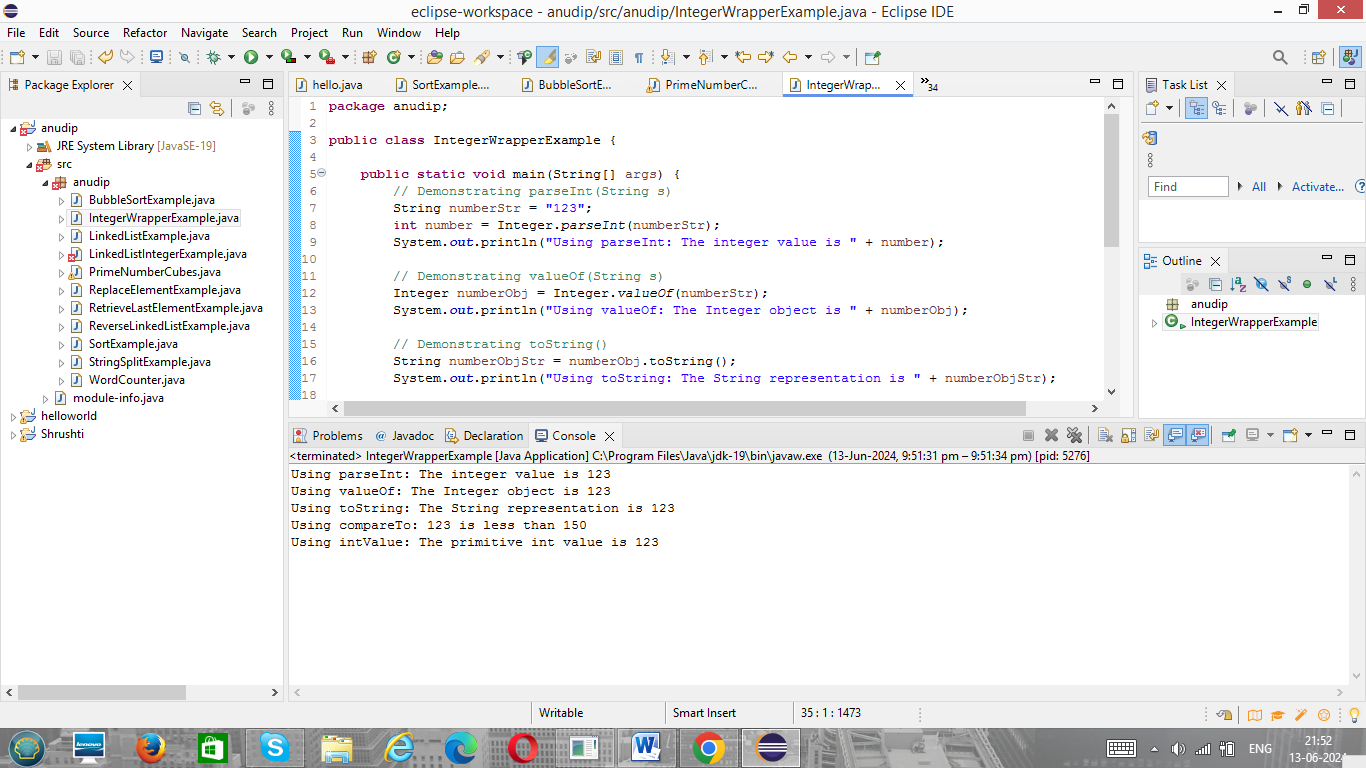
double primitiveDouble = numberObj.doubleValue();

System.out.println("Using doubleValue: The primitive double value is " + primitiveDouble);

}

}

**Output:**



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

**Code:**

public class FloatWrapperExample {

public static void main(String[] args) {

// Demonstrating parseFloat(String s)

String floatStr = "123.45";

float number = Float.parseFloat(floatStr);

System.out.println("Using parseFloat: The float value is " + number);

// Demonstrating valueOf(String s)

Float numberObj = Float.valueOf(floatStr);

System.out.println("Using valueOf: The Float object is " + numberObj);

// Demonstrating toString()

String numberObjStr = numberObj.toString();

System.out.println("Using toString: The String representation is " + numberObjStr);

// Demonstrating compareTo(Float anotherFloat)

Float anotherNumberObj = 150.75f;

int comparisonResult = numberObj.compareTo(anotherNumberObj);

if (comparisonResult < 0) {

System.out.println("Using compareTo: " + numberObj + " is less than " + anotherNumberObj);

} else if (comparisonResult > 0) {

System.out.println("Using compareTo: " + numberObj + " is greater than " + anotherNumberObj);

} else {

System.out.println("Using compareTo: " + numberObj + " is equal to " + anotherNumberObj);

}

// Demonstrating floatValue()

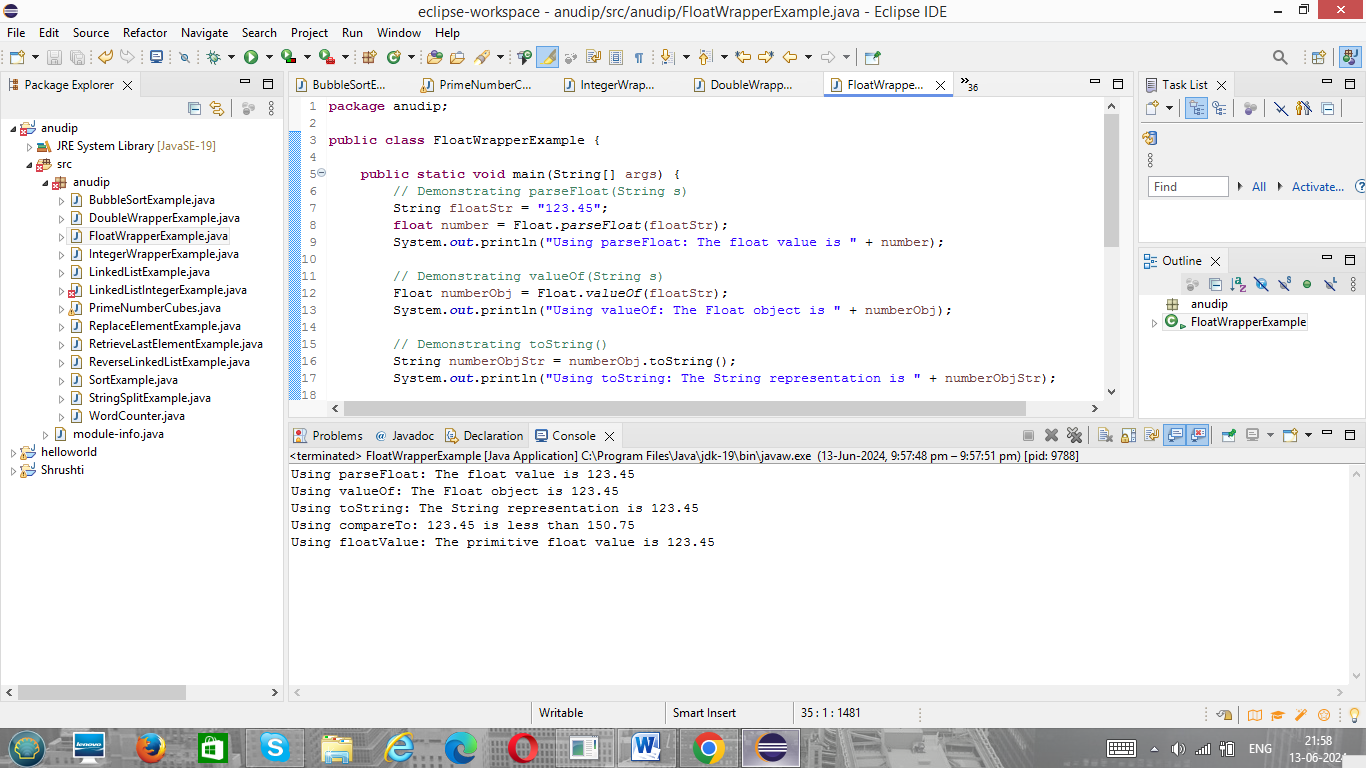
float primitiveFloat = numberObj.floatValue();

System.out.println("Using floatValue: The primitive float value is " + primitiveFloat);

}

}

**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.**

**Code:**

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import java.util.Scanner;

public class EmailValidator {

public static void main(String[] args) {

// Define the regex pattern for a valid email address

String emailRegex = "^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,6}$";

// Compile the regex into a Pattern object

Pattern pattern = Pattern.compile(emailRegex);

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

Scanner scanner = new Scanner(System.in);

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

String email = scanner.nextLine();

// Match the email against the pattern

Matcher matcher = pattern.matcher(email);

// Check if the email matches the pattern

if (matcher.matches()) {

System.out.println("The email address is valid.");

} else {

System.out.println("The email address is invalid.");

}

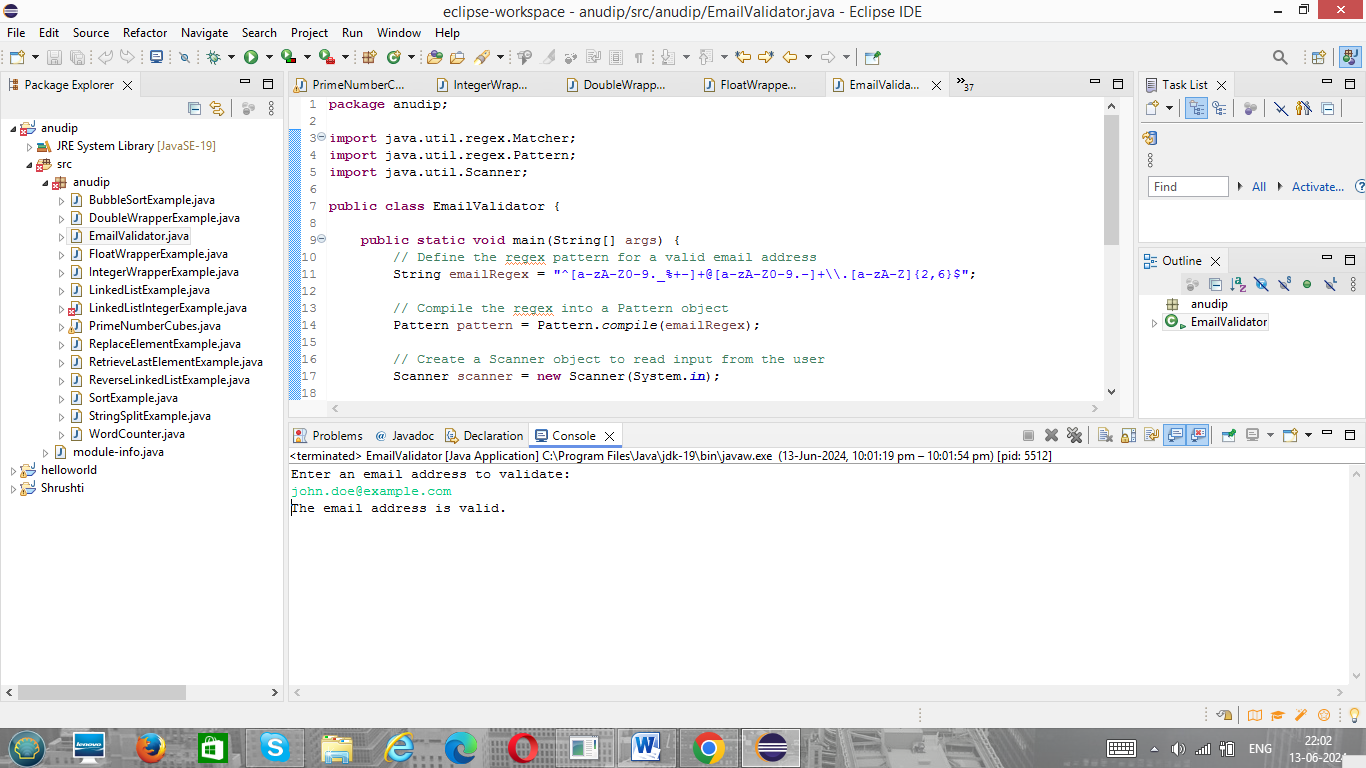
// Close the scanner

scanner.close();

}

}

**Output:**



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

**Code:**

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import java.util.Scanner;

public class PhoneNumberValidator {

public static void main(String[] args) {

// Define the regex pattern for a valid phone number

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

// Compile the regex into a Pattern object

Pattern pattern = Pattern.compile(phoneRegex);

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

Scanner scanner = new Scanner(System.in);

System.out.println("Enter a phone number to validate (format: (xxx) xxx-xxxx):");

String phoneNumber = scanner.nextLine();

// Match the phone number against the pattern

Matcher matcher = pattern.matcher(phoneNumber);

// Check if the phone number matches the pattern

if (matcher.matches()) {

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

} else {

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

}

// Close the scanner

scanner.close();

}

}

**Output:**

