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** mypackage;

**import** java.util.Arrays;

**public** **class** SortedExample {

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

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

String[] strArr = {"c", "O", "I","P","U"};

Arrays.*sort*(intArr);

Arrays.*sort*(strArr);

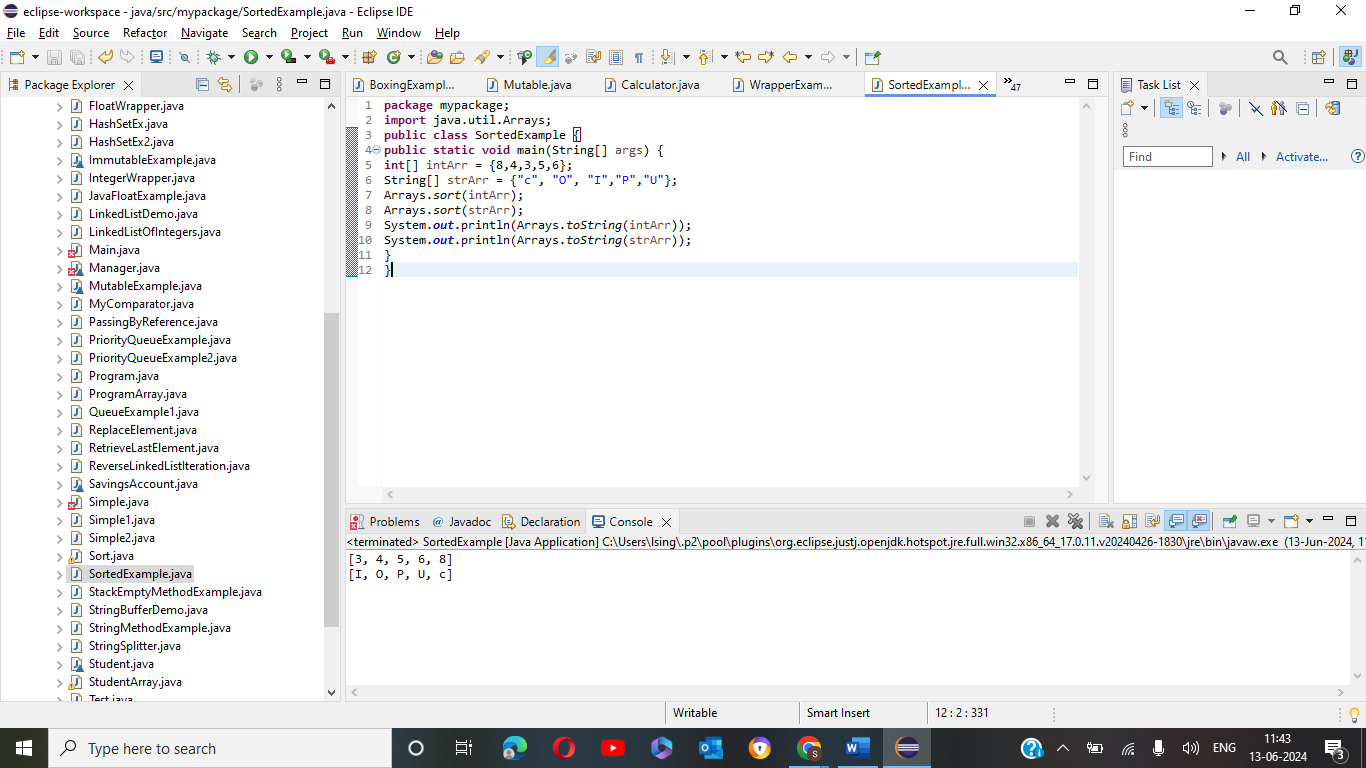
System.***out***.println(Arrays.*toString*(intArr));

System.***out***.println(Arrays.*toString*(strArr));

}

}

Program-



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

Code-

**package** mypackage;

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

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

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

*bubbleSort*(numbers);

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

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

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

}

}

**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 two elements were swapped in the inner loop, then the array is already sorted

**if** (!swapped) **break**;

}

}

}

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** mypackage;

**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 10 elements into the array

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

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

numbers[i] = scanner.nextInt();

}

// Process and print the cube of prime numbers

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

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

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

System.***out***.println(num + " cubed is " + (num \* num \* 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**;

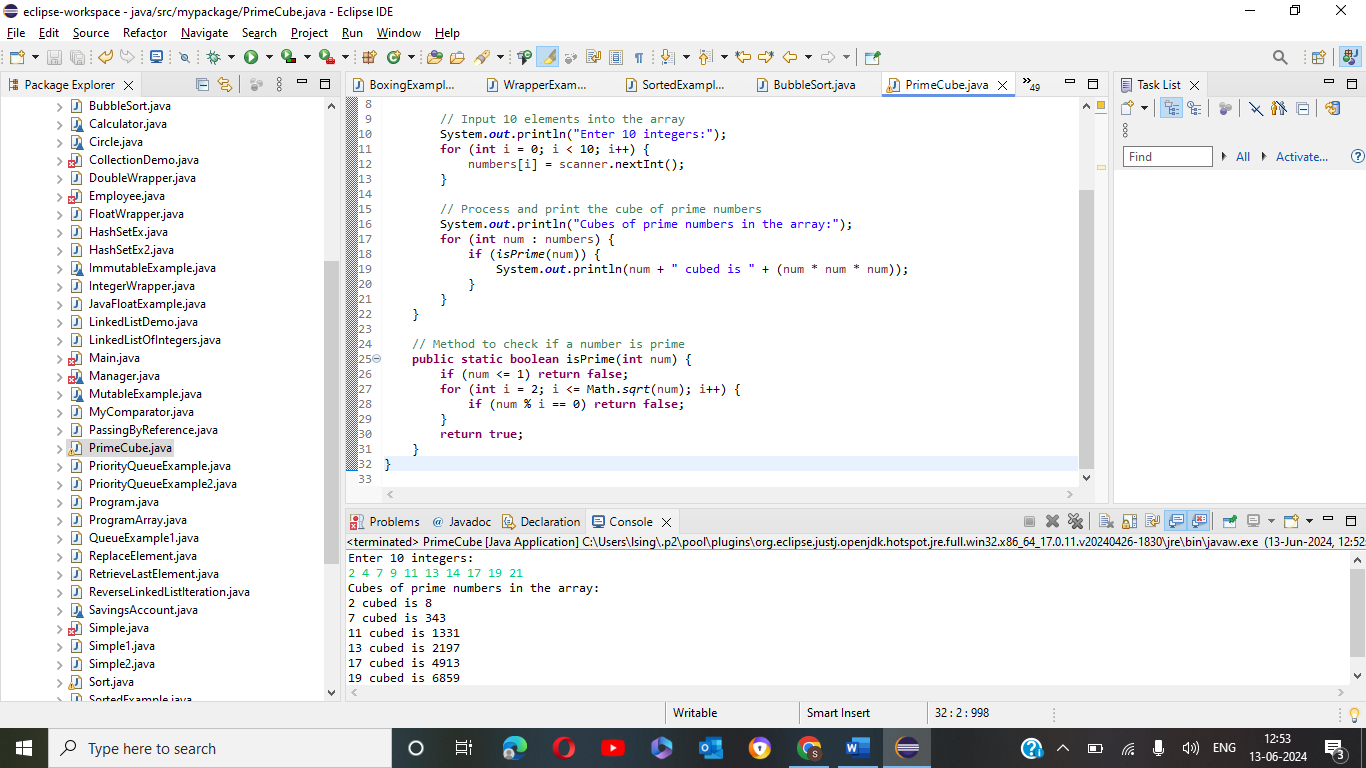
}

**return** **true**;

}

}

Program-



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

Code-

**package** mypackage;

**public** **class** IntegerWrapperClassMethod {

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

String numberStr = "123";

**int** parsedNumber = Integer.*parseInt*(numberStr);

System.***out***.println("Parsed number: " + parsedNumber);

**int** number = 456;

String numberStr2 = Integer.*toString*(number);

System.***out***.println("String representation of number: " + numberStr2);

String numberStr3 = "789";

Integer integerObject = Integer.*valueOf*(numberStr3);

System.***out***.println("Integer object from string: " + integerObject);

Integer intObj1 = Integer.*valueOf*(10);

Integer intObj2 = Integer.*valueOf*(20);

**int** comparisonResult = intObj1.compareTo(intObj2);

System.***out***.println("Comparison result (10 compared to 20): " + comparisonResult);

Integer integerObject2 = Integer.*valueOf*(42);

**int** intValue = integerObject2.intValue();

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

}

}

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** mypackage;

**public** **class** DoubleWrapperClassMethods {

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

String doubleStr = "123.45";

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

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

**double** number = 456.78;

String doubleStr2 = Double.*toString*(number);

System.***out***.println("String representation of double: " + doubleStr2);

String doubleStr3 = "789.01";

Double doubleObject = Double.*valueOf*(doubleStr3);

System.***out***.println("Double object from string: " + doubleObject);

Double doubleObj1 = Double.*valueOf*(10.5);

Double doubleObj2 = Double.*valueOf*(20.5);

**int** comparisonResult = doubleObj1.compareTo(doubleObj2);

System.***out***.println("Comparison result (10.5 compared to 20.5): " + comparisonResult);

Double doubleObj3 = Double.*valueOf*(Double.***NaN***);

**boolean** isNaN = doubleObj3.isNaN();

System.***out***.println("Is NaN: " + isNaN); // Output: Is NaN: true

}

}

Program-

A screenshot of a computer

Description automatically generated

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

Code-

**package** mypackage;

**public** **class** FloatWrapperClassMethods {

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

String floatStr = "123.45";

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

System.***out***.println("Parsed float: " + parsedFloat);

**float** number = 456.78f;

String floatStr2 = Float.*toString*(number);

System.***out***.println("String representation of float: " + floatStr2);

String floatStr3 = "789.01";

Float floatObject = Float.*valueOf*(floatStr3);

System.***out***.println("Float object from string: " + floatObject);

Float floatObj1 = Float.*valueOf*(10.5f);

Float floatObj2 = Float.*valueOf*(20.5f);

**int** comparisonResult = floatObj1.compareTo(floatObj2);

System.***out***.println("Comparison result (10.5 compared to 20.5): " + comparisonResult);

Float floatObj3 = Float.*valueOf*(Float.***NaN***);

**boolean** isNaN = floatObj3.isNaN();

System.***out***.println("Is NaN: " + isNaN); // Output: Is NaN: true

}

}

Program-

A screenshot of a computer

Description automatically generated

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** mypackage;

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

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

**import** java.util.Scanner;

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

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

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

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

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

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

String email = scanner.nextLine();

Matcher matcher = pattern.matcher(email);

**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();

}

}

Program-

A screenshot of a computer

Description automatically generated

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

Code-

**package** mypackage;

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

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

**import** java.util.Scanner;

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

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

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

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

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

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

String phoneNumber = scanner.nextLine();

Matcher matcher = pattern.matcher(phoneNumber);

**if** (matcher.matches()) {

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

} **else** {

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

}

scanner.close();

}

}

Program-

