**Lab 6**

**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**

**Program**

**package** lab6;

**import** java.util.Arrays;

**public** **class** SortExample {

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

// Sorting integers

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

Arrays.*sort*(numbers); // Sorting integers in ascending order

// Displaying sorted integers

System.***out***.print("Sorted numbers in ascending order: ");

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

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

}

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

// Sorting alphabetical strings

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

Arrays.*sort*(strings); // Sorting strings in ascending order

// Displaying sorted strings

System.***out***.print("Sorted strings in ascending order: ");

**for** (String str : strings) {

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

}

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

}

}

**Output**

****

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

**Program**

**package** lab6;

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

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

// Array to be sorted

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

// Call the bubbleSort method

*bubbleSort*(array);

// Display the sorted array

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

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

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

}

}

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

**int** n = array.length;

**boolean** swapped;

// Perform n-1 passes

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

swapped = **false**;

// Perform pairwise comparisons and swap if needed

**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 sorted

**if** (!swapped) {

**break**;

}

}

}

}

**Output**

**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**

**Program**

**package** lab6;

**import** java.util.Scanner;

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

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

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

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

// Input array elements

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** i = 0; i < 10; i++) {

**if** (*isPrime*(array[i])) {

System.***out***.println(array[i] + "^3 = " + *cube*(array[i]));

}

}

scanner.close();

}

// Method to check if a number is prime

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

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

**return** **false**;

}

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

**if** (number % i == 0) {

**return** **false**;

}

}

**return** **true**;

}

// Method to calculate the cube of a number

**public** **static** **int** cube(**int** number) {

**return** number \* number \* number;

}

}

**Output**

**A screenshot of a computer

Description automatically generated**

**4.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) {

// 1. Integer.parseInt(String s)

String numberStr = "123";

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

System.***out***.println("Integer.parseInt(\"" + numberStr + "\") = " + number);

// 2. Integer.valueOf(String s)

Integer numberObj = Integer.*valueOf*(numberStr);

System.***out***.println("Integer.valueOf(\"" + numberStr + "\") = " + numberObj);

// 3. Integer.compare(int x, int y)

**int** num1 = 10;

**int** num2 = 20;

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

System.***out***.println("Integer.compare(" + num1 + ", " + num2 + ") = " + comparisonResult);

// 4. Integer.toString(int i)

**int** num3 = 456;

String numStr = Integer.*toString*(num3);

System.***out***.println("Integer.toString(" + num3 + ") = " + numStr);

// 5. Integer.bitCount(int i)

**int** num4 = 29; // Binary: 11101

**int** bitCount = Integer.*bitCount*(num4);

System.***out***.println("Integer.bitCount(" + num4 + ") = " + bitCount);

}

}

**Output**

**A screenshot of a computer program

Description automatically generated**

**5.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) {

// 1. Double.parseDouble(String s)

String doubleStr = "123.45";

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

System.***out***.println("Double.parseDouble(\"" + doubleStr + "\") = " + number);

// 2. Double.valueOf(String s)

Double numberObj = Double.*valueOf*(doubleStr);

System.***out***.println("Double.valueOf(\"" + doubleStr + "\") = " + numberObj);

// 3. Double.isNaN(double v)

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

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

System.***out***.println("Double.isNaN(Double.NaN) = " + isNan);

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

**double** num1 = 10.5;

**double** num2 = 20.5;

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

System.***out***.println("Double.compare(" + num1 + ", " + num2 + ") = " + comparisonResult);

// 5. Double.toString(double d)

**double** num3 = 456.78;

String numStr = Double.*toString*(num3);

System.***out***.println("Double.toString(" + num3 + ") = " + numStr);

}

}

**Output**

**A screenshot of a computer program

Description automatically generated**

**6.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) {

// 1. Float.parseFloat(String s)

String floatStr = "123.45f";

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

System.***out***.println("Float.parseFloat(\"" + floatStr + "\") = " + number);

// 2. Float.valueOf(String s)

Float numberObj = Float.*valueOf*(floatStr);

System.***out***.println("Float.valueOf(\"" + floatStr + "\") = " + numberObj);

// 3. Float.isNaN(float v)

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

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

System.***out***.println("Float.isNaN(Float.NaN) = " + isNan);

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

**float** num1 = 10.5f;

**float** num2 = 20.5f;

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

System.***out***.println("Float.compare(" + num1 + ", " + num2 + ") = " + comparisonResult);

// 5. Float.toString(float f)

**float** num3 = 456.78f;

String numStr = Float.*toString*(num3);

System.***out***.println("Float.toString(" + num3 + ") = " + numStr);

}

}

**Output**

**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.**

**Program**

**package** lab6;

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

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

**import** java.util.Scanner;

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

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

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

// Regular expression for validating email addresses

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

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

// Input email address

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

String email = scanner.nextLine();

// Validate email address

Matcher matcher = emailPattern.matcher(email);

**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**

**A white background with black text

Description automatically generated**

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

**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***);

// Regular expression for validating phone numbers

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

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

// Input phone number

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

String phoneNumber = scanner.nextLine();

// Validate phone number

Matcher matcher = phonePattern.matcher(phoneNumber);

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

System.***out***.println(phoneNumber + " is a valid phone number.");

} **else** {

System.***out***.println(phoneNumber + " is not a valid phone number.");

}

scanner.close();

}

}

**Output**

**A white background with black text

Description automatically generated**