Lab 6

Aditya kamble

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.

**-**

**package** box;

**import** java.util.Arrays;

**public** **class** sorts {

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

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

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

Arrays.*sort*(intArray);

Arrays.*sort*(stringArray);

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

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

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

}

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

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

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

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

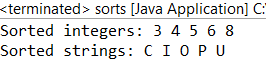
}

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

}

}

**OUTPUT**



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

**-**

**package** box;

**public** **class** bu {

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

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

*bubbleSort*(array);

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

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

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]) {

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

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

array[j + 1] = temp;

swapped = **true**;

}

}

**if** (!swapped) {

**break**;

}

}

}

}

**OUTPUT**



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

**-**

**package** box;

**import** java.util.Scanner;

**public** **class** asc {

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

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

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

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

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

array[i] = scanner.nextInt();

}

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

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

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

**int** cube = array[i] \* array[i] \* array[i];

System.***out***.println("Prime number: " + array[i] + ", Cube: " + cube);

}

}

scanner.close();

}

**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**;

}

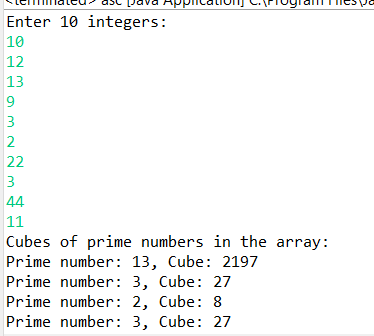
}

**return** **true**;

}

}

**OUTPUT**



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

**-**

**package** box;

**public** **class** rap {

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

String numStr = "123";

**int** num = Integer.*parseInt*(numStr);

System.***out***.println("Method 1: parseInt(String s) - Converted string to int: " + num);

**int** value = 456;

String strValue = Integer.*toString*(value);

System.***out***.println("Method 2: toString(int i) - Converted int to string: " + strValue);

Integer intObject = **new** ~~Integer~~(789);

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

System.***out***.println("Method 3: intValue() - Converted Integer object to int: " + intValue);

Integer num1 = 100;

Integer num2 = 200;

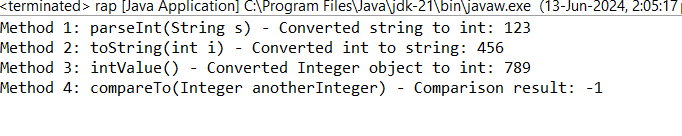
**int** result = num1.compareTo(num2);

System.***out***.println("Method 4: compareTo(Integer anotherInteger) - Comparison result: " + result);

}

}

**OUTPUT**



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

**-**

**package** box;

**public** **class** DO {

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

String doubleString = "12.34";

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

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

**double** primitiveDouble = doubleObject.doubleValue();

System.***out***.println("Primitive double value: " + primitiveDouble);

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

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

System.***out***.println("Is NaN: " + isNan);

**double** value1 = 15.5;

**double** value2 = 14.5;

**int** comparisonResult = Double.*compare*(value1, value2);

System.***out***.println("Comparison result : " + comparisonResult);

value1 = 10.5;

value2 = 20.5;

comparisonResult = Double.*compare*(value1, value2);

System.***out***.println("Comparison result: " + comparisonResult);

String anotherDoubleString = "45.67";

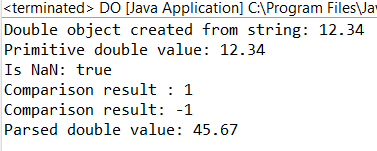
**double** parsedDouble = Double.*parseDouble*(anotherDoubleString);

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

}

}

**OUTPUT**



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

**-**

**package** box;

**public** **class** F {

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

String floatString = "12.34";

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

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

**float** primitiveFloat = floatObject.floatValue();

System.***out***.println("Primitive float value: " + primitiveFloat);

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

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

System.***out***.println("Is NaN: " + isNan);

**float** value1 = 15.5f;

**float** value2 = 15.5f;

**int** comparisonResult = Float.*compare*(value1, value2);

System.***out***.println("Comparison result : " + comparisonResult);

value1 = 10.5f;

value2 = 20.5f;

comparisonResult = Float.*compare*(value1, value2);

System.***out***.println("Comparison result : " + comparisonResult);

String anotherFloatString = "45.67";

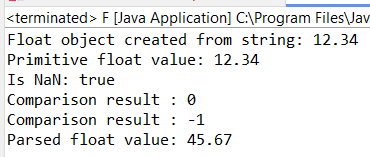
**float** parsedFloat = Float.*parseFloat*(anotherFloatString);

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

}

}

**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

**-**

**package** emp;

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

**public** **class** emps {

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

String email1 = "[veer@gmail.com](mailto:veer@gmail.com)";

String email2 = "rajgmail.com";

System.***out***.println("email1 = " + *validateEmail*(email1));

System.***out***.println("email2 = " + *validateEmail*(email2));

}

**public** **static** String validateEmail (String email) {

**if**(email == **null** || email.isEmpty()) {

**return** "Invalid";

}

String emailRegex = "^[a-zA-Z0-9\_+&\*-]+(?:\\." + "[a-zA-Z0-9\_+&\*-]+)\*@" +"(?:[a-zA-Z0-9-]+\\.)+[a-zA-Z]{2,7}$";

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

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

**return** "valid";

}

**else** {

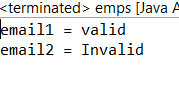
**return** "Invalid";

}

}

}

**OUTPUT**



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

**-**

**package** cal;

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

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

**public** **class** mobile {

**private** **static** **final** String ***PHONE\_REGEX*** = "^\\(\\d{3}\\) \\d{3}-\\d{4}$";

**public** **static** **boolean** isValidPhoneNumber(String phoneNumber) {

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

Matcher matcher = pattern.matcher(phoneNumber);

**return** matcher.matches();

}

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

String[] phoneNumbers = {

"(123) 456-7890",

"(123) 456-789",

"123-456-7890",

"(123)456-7890",

"(123) 456-78901"

};

**for** (String phoneNumber : phoneNumbers) {

**if** (*isValidPhoneNumber*(phoneNumber)) {

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

} **else** {

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

}

}

}

}

**OUTPUT**

