1. **Write the 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** myPackage;

**import** java.util.\*;

**public** **class** Sort\_An\_Array {

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

// **TODO** Auto-generated method stub

**int** arr1[]= {8,4,3,5,6};

Arrays.*sort*(arr1);

**for**(**int** i:arr1 ) {

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

}

String arr2[]= {"C", "O","I","P","U",};

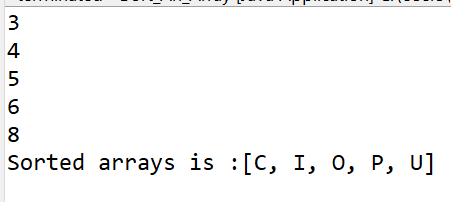
Arrays.*sort*(arr2);

System.***out***.println("Sorted arrays is :"+Arrays.*toString*(arr2));

}

}

**Output:**



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

**Program:**

**package** myPackage;

**public** **class** Bubble\_Sort {

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

**int** n = arr.length;

**boolean** swapped;

// Loop through each element in the array

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

swapped = **false**;

// Loop through the array from the beginning to n-i-1

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

// Swap if the element found is greater than the next element

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

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

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

arr[j + 1] = temp;

swapped = **true**;

}

}

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

**if** (!swapped) {

**break**;

}

}

}

// Method to print the array

**public** **static** **void** printArray(**int**[] arr) {

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

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

}

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

}

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

// **TODO** Auto-generated method stub

**int**[] arr = {64, 34, 25, 12, 22, 11, 90};

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

*printArray*(arr);

*bubbleSort*(arr);

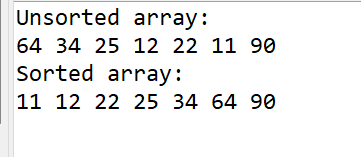
System.***out***.println("Sorted array:");

*printArray*(arr);

}

}

**Output:**



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

**Program:**

**package** myPackage;

**import** java.util.Scanner;

**public** **class** Cube\_Of\_Prime\_No {

**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 compute the cube of a number

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

**return** num \* num \* num;

}

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

// **TODO** Auto-generated method stub

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

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

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

// Input 10 elements into the array

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

arr[i] = scanner.nextInt();

}

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

// Check each element if it is prime, then print its cube if it is

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

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

System.***out***.println("Cube of " + arr[i] + " is " + *cube*(arr[i]));

}

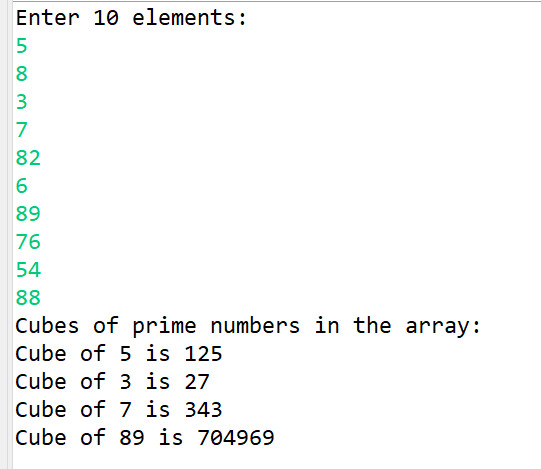
}

scanner.close();

}

}

**Output:**



**4.Write a java program to implement integer wrapper class methods.(any 3 methods)**

**Program:**

**package** myPackage;

**public** **class** Integer\_Wraper\_With\_Three\_Methods {

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

// **TODO** Auto-generated method stub

// Method 1: parseInt()

String numberString = "123";

**int** parsedInt = Integer.*parseInt*(numberString);

System.***out***.println("1. parseInt() result: " + parsedInt);

// Method 2: valueOf()

Integer valueOfInt = Integer.*valueOf*(456);

System.***out***.println("2. valueOf() result: " + valueOfInt);

// Method 3: compareTo()

Integer num1 = 10;

Integer num2 = 20;

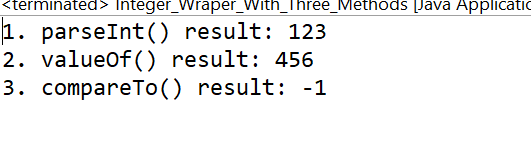
**int** compareResult = num1.compareTo(num2);

System.***out***.println("3. compareTo() result: " + compareResult);

}

}

**Output:**



**5. Write a java program to implement double wrapper class methods.(any 3 methods)**

**Program:**

**package** myPackage;

**public** **class** Double\_Wraper\_Class\_With\_Methods {

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

// **TODO** Auto-generated method stub

// Method 1: parseDouble()

String doubleString = "3.14159";

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

System.***out***.println("1. parseDouble() result: " + parsedDouble);

// Method 2: valueOf()

Double valueOfDouble = Double.*valueOf*(2.71828);

System.***out***.println("2. valueOf() result: " + valueOfDouble);

// Method 3: isInfinite()

Double infiniteDouble = Double.***POSITIVE\_INFINITY***;

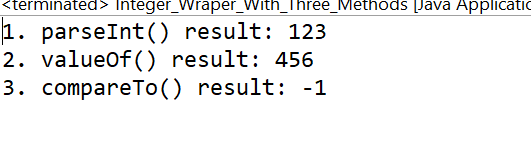
**boolean** isInfinite = infiniteDouble.isInfinite();

System.***out***.println("3. isInfinite() result: " + isInfinite);

}

}

**Output:**



**6.     Write a java program to implement float wrapper class methods.(any 3 methods)**

**package** myPackage;

**public** **class** Float\_Wrapper\_Class\_With\_Methods {

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

// **TODO** Auto-generated method stub

// Method 1: parseFloat()

String floatString = "3.14";

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

System.***out***.println("1. parseFloat() result: " + parsedFloat);

// Method 2: valueOf()

Float valueOfFloat = Float.*valueOf*(2.718f);

System.***out***.println("2. valueOf() result: " + valueOfFloat);

// Method 3: isNaN()

Float nanFloat = Float.***NaN***;

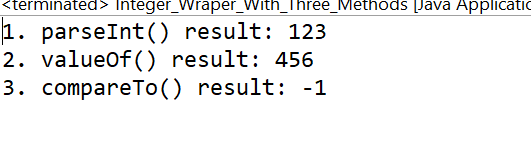
**boolean** isNaN = nanFloat.isNaN();

System.***out***.println("3. isNaN() result: " + isNaN);

}

}

**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.**

**Program:**

**package** myPackage;

**import** java.util.Scanner;

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

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

**public** **class** Email\_Validator {

// Method to validate email using regular expression

**public** **static** **boolean** isValidEmail(String email) {

// Regular expression for validating email

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

// Compile the regex

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

// Match the email with the regex

Matcher matcher = pattern.matcher(email);

// Return whether the email matched the regex

**return** matcher.matches();

}

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

// **TODO** Auto-generated method stub

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

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

String email = scanner.nextLine();

**if** (*isValidEmail*(email)) {

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

} **else** {

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

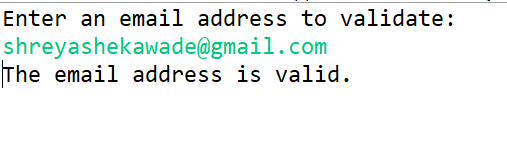
}

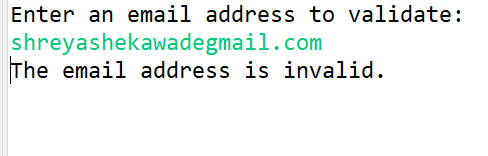
scanner.close();

}

}

**Output:**

****

****

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

**Program:**

**package myPackage;**

**import java.util.Scanner;**

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

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

**public class Phone\_No\_Validetor {**

**// Method to validate phone number using regular expression**

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

**// Regular expression for validating phone number**

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

**// Compile the regex**

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

**// Match the phone number with the regex**

**Matcher matcher = pattern.matcher(phoneNumber);**

**// Return whether the phone number matched the regex**

**return matcher.matches();**

**}**

**// Main method to test the phone number validation**

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

**// TODO Auto-generated method stub**

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

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

**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();**

**}**

**}**

**Output:**

