1. Program: Find out duplicate number between 1 to N numbers.

Description: You have got a range of numbers between 1 to N, where one of the number is repeated. You need to write a program to find out the duplicate number.

**package** CoreXJava;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** DuplicateNumber {

**public** **int** findDuplicateNumber(List<Integer> numbers){

**int** highestNumber = numbers.size() - 1;

**int** total = getSum(numbers);

**int** duplicate = total - (highestNumber\*(highestNumber+1)/2);

**return** duplicate;

}

**public** **int** getSum(List<Integer> numbers){

**int** sum = 0;

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

sum += num;

}

**return** sum;

}

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

List<Integer> numbers = **new** ArrayList<Integer>();

**for**(**int** i=1;i<30;i++){

numbers.add(i);

}

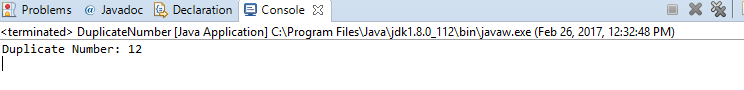
//add duplicate number into the list

numbers.add(12);

DuplicateNumber dn = **new** DuplicateNumber();

System.***out***.println("Duplicate Number: "+dn.findDuplicateNumber(numbers));

}

}

1. Program: Write a program to reverse a number.

Description: Write a program to reverse a number using numeric operations. Below example shows how to reverse a number using numeric operations.

**package** CoreXJava;

**import** java.util.Scanner;

**public** **class** Palindrome

{

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

{

**int** n, a = 0,x;

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

System.***out***.print("Enter any number:");

n = s.nextInt();

**while**(n > 0)

{

x = n % 10;

a = a \* 10 + x;

n = n / 10;

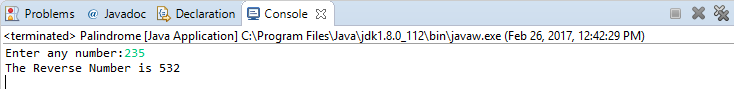
}

System.***out***.println("The Reverse Number is "+a);

s.close();

}

}



1. Program: Write a program to find perfect number or not.

Description: A perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself. Equivalently, a perfect number is a number that is half the sum of all of its positive divisors excluding the number itself. Equivalently, a perfect number is a number that is half the sum of all of its positive divisors.

The first perfect number is 6, because 1, 2 and 3 are its proper positive divisors, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors:

**package** CoreXJava;

**import** java.util.Scanner;

**public** **class** Perfect

{

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

{

**int** n, sum = 0;

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

System.***out***.print("Enter any integer you want to check:");

n = s.nextInt();

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

{

**if**(n % i == 0)

{

sum = sum + i;

}

}

**if**(sum == n)

{

System.***out***.println("Given number is Perfect");

}

**else**

{

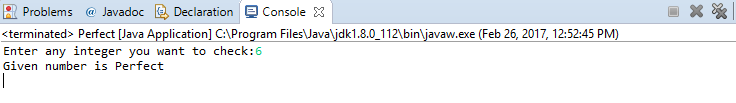
System.***out***.println("Given number is not Perfect");

}

s.close();

}

}



1. Program: Write a program to implement ArrayList.

Description: Write a program to implement your own ArrayList class. It should contain add(), get(), remove(), size() methods. Use dynamic array logic. It should increase its size when it reaches threshold.

**package** CoreXJava;

**import** java.util.Arrays;

**import** java.util.Scanner;

**public** **class** MyArrayList {

**private** Object[] myStore;

**private** **int** actSize = 0;

**public** MyArrayList(){

myStore = **new** Object[10];

}

**public** Object get(**int** index){

**if**(index < actSize){

**return** myStore[index];

} **else** {

**throw** **new** ArrayIndexOutOfBoundsException();

}

}

**public** **void** add(Object obj){

**if**(myStore.length-actSize <= 5){

increaseListSize();

}

myStore[actSize++] = obj;

}

**public** Object remove(**int** index){

**if**(index < actSize){

Object obj = myStore[index];

myStore[index] = **null**;

**int** tmp = index;

**while**(tmp < actSize){

myStore[tmp] = myStore[tmp+1];

myStore[tmp+1] = **null**;

tmp++;

}

actSize--;

**return** obj;

} **else** {

**throw** **new** ArrayIndexOutOfBoundsException();

}

}

**public** **int** size(){

**return** actSize;

}

**private** **void** increaseListSize(){

myStore = Arrays.*copyOf*(myStore, myStore.length\*2);

System.***out***.println("New length: "+myStore.length);

}

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

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

MyArrayList mal = **new** MyArrayList();

mal.add(sc.nextInt());

mal.add(sc.nextInt());

mal.add(sc.nextInt());

mal.add(sc.nextInt());

mal.add(sc.nextInt());

**for**(**int** i=0;i<mal.size();i++){

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

}

mal.add(sc.nextInt());

System.***out***.println("Element at Index 5="+mal.get(5));

System.***out***.println("List size: "+mal.size());

System.***out***.println("Removing element at index 2: "+mal.remove(2));

**for**(**int** i=0;i<mal.size();i++){

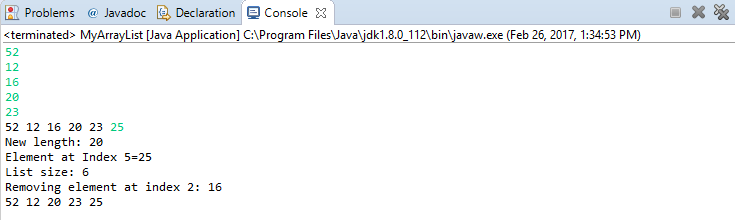
System.***out***.print(mal.get(i)+" ");

sc.close();

}

}

}



1. Program to find second highest number in an integer array without sorting the elements.

**package** CoreXJava;

**class** SecondLargestNumber{

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

{

**int** numbers[] = {6,3,37,12,46,5,64,21};

**int** highest = 0;

**int** second\_highest = 0;

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

**if**(highest < n){

second\_highest = highest;

highest =n;

} **else** **if**(second\_highest < n){

second\_highest = n;

}

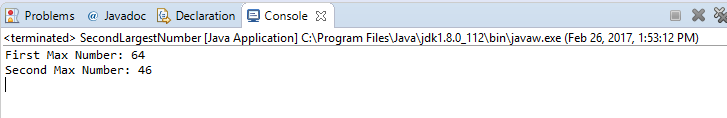
}

System.***out***.println("First Max Number: "+highest);

System.***out***.println("Second Max Number: "+second\_highest);

}

}



6. Java Program for Fibonacci series without using recursive function concept.

**package** CoreXJava;

**class** FibonacciExample

{

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

{

**int** n1=0,n2=1,n3,i,count=10;

System.***out***.print(n1+" "+n2);

**for**(i=2;i<count;++i)

{

n3=n1+n2;

System.***out***.print(" "+n3);

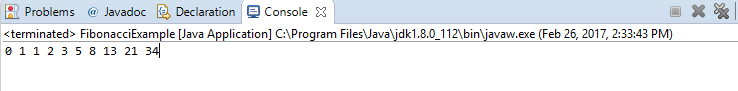
n1=n2;

n2=n3;

}

}

}



1. Program: Write a program to find sum of each digit in the given number using recursion.

Description: Below example shows how to find out sum of each digit in the given number using recursion logic. For example, if the number is 259, then the sum should be 2+5+9 = 16

**package** CoreXJava;

**import** java.util.Scanner;

**public** **class** DigitSum

{

**int** sum = 0;

**public** **int** getNumberSum(**int** number)

{

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

**return** sum;

} **else** {

sum += (number%10);

getNumberSum(number/10);

}

**return** sum;

}

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

DigitSum mns = **new** DigitSum();

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

System.***out***.println("Enter a Number");

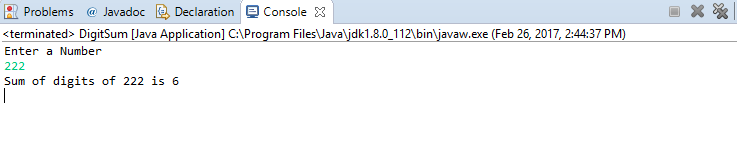
**int** num = sc.nextInt();

System.***out***.println("Sum of digits of "+num+" is "+mns.getNumberSum(num));

sc.close();

}

}



8. Program: Write a program to find the sum of the first 1000 prime numbers.

Description: Write a program to find the sum of the first 1000 prime numbers.

**package** CoreXJava;

**public** **class** PrimeAdd {

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

**int** number = 2;

**int** count = 0;

**long** sum = 0;

**while**(count < 1000){

**if**(*isPrimeNumber*(number)){

sum += number;

count++;

}

number++;

}

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

}

**private** **static** **boolean** isPrimeNumber(**int** number){

**for**(**int** i=2; i<=number/2; i++){

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

**return** **false**;

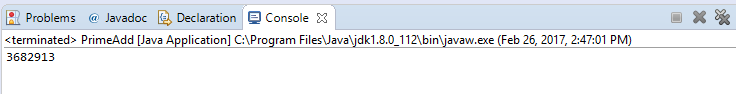
}

}

**return** **true**;

}

}



1. Program: How to swap two numbers without using temporary variable?

Description: Write a program to swap or exchange two numbers. You should not use any temporary or third variable to swap.

**package** CoreXJava;

**public** **class** SwapWithoutTemp

{

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

{

**int** num1 = 74;

**int** num2 = 62;

System.***out***.println("Before Swapping");

System.***out***.println("Value of num1 is :" + num1);

System.***out***.println("Value of num2 is :" +num2);

num1 = num1 + num2;

num2 = num1 - num2;

num1 = num1 - num2;

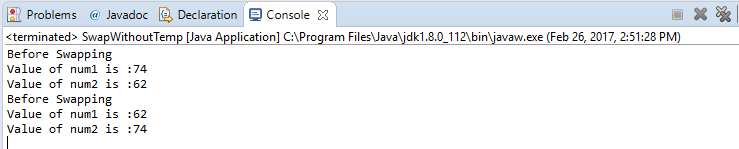
System.***out***.println("Before Swapping");

System.***out***.println("Value of num1 is :" + num1);

System.***out***.println("Value of num2 is :" +num2);

}

}



1. Write a Program to find top 2 maximum numbers in array.

You should not use any sorting functions. You should iterate the array only once. You should not use any kind of collections in java.

**package** CoreXJava;

**public** **class** TwoMaxNumbers {

**public** **void** printTwoMaxNumbers(**int**[] nums){

**int** maxOne = 0;

**int** maxTwo = 0;

**for**(**int** n:nums){

**if**(maxOne < n){

maxTwo = maxOne;

maxOne =n;

} **else** **if**(maxTwo < n){

maxTwo = n;

}

}

System.***out***.println("First Max Number: "+maxOne);

System.***out***.println("Second Max Number: "+maxTwo);

}

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

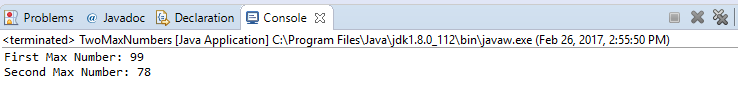
**int** num[] = {5,34,78,2,45,1,99,23};

TwoMaxNumbers tmn = **new** TwoMaxNumbers();

tmn.printTwoMaxNumbers(num);

}

}



1. Program: Floyd Triangle 1 2 3 4 5 6 7 8 9 10

**package** CoreXJava;

**import** java.util.Scanner;

**class** FloydTriangle

{

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

{

**int** n, num = 1, c, d;

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

System.***out***.println("Enter the number of rows of floyd's triangle you want");

n = in.nextInt();

System.***out***.println("Floyd's triangle :-");

**for** ( c = 1 ; c <= n ; c++ )

{

**for** ( d = 1 ; d <= c ; d++ )

{

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

num++;

}

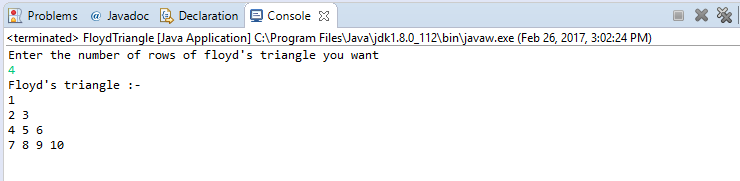
System.***out***.println();

in.close();

}

}

}



1. How to count occurrence of a given character in a String?

**package** CoreXJava;

**import** java.util.HashMap;

**public** **class** CountString {

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

String str = "BigDataHadoop";

HashMap<Character, Integer> map = **new** HashMap<>();

**for** (**char** ch : str.toCharArray()) {

**if** (map.containsKey(ch)) {

**int** val = map.get(ch);

map.put(ch, val + 1);

} **else** {

map.put(ch, 1);

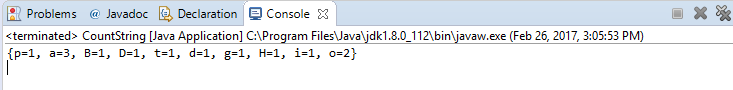
}

}

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

}

}



1. Program: Write a program to check whether accepted number is Magic number.

Description:

A Magic number is a number whose sum of digits eventually leads to 1. Example#1: 19 ; 1+9 =10 ; 1+0 = 1. Hence a magic number. Example#2: 226; 2+2+6=10; 1+0 =1. Hence a magic number. Example#3: 874; 8+7+4=19; 1+9=10; 1+0=1. Hence a magic number.

**package** CoreXJava;

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

**public** **class** MagicNumberCheck

{

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

{

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

System.***out***.println("Enter the number to be checked.");

**int** n=ob.nextInt();

**int** sum=0,num=n;

**while**(num>9)

{

sum=num;**int** s=0;

**while**(sum!=0)

{

s=s+(sum%10);

sum=sum/10;

}

num=s;

}

**if**(num==1)

{

System.***out***.println(n+" is a Magic Number.");

}

**else**

{

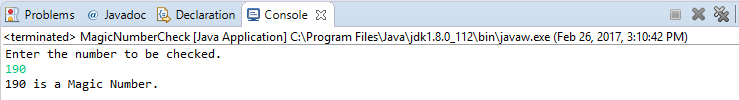
System.***out***.println(n+" is not a Magic Number.");

}

ob.close();

}

}



1. Program: Write a program How to check if a number is binary.

Description:

For this question, you need to write a function which will accept an integer and return true if it contains only 0 and 1 e.g. if input is 123 then your function will return false, for 101 it should return true.

**package** CoreXJava;

**public** **class** BinaryCheck {

**public** **boolean** isBinaryNumber(**int** binary){

**boolean** status = **true**;

**while**(**true**){

**if**(binary == 0){

**break**;

} **else** {

**int** tmp = binary%10;

**if**(tmp > 1){

status = **false**;

**break**;

}

binary = binary/10;

}

}

**return** status;

}

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

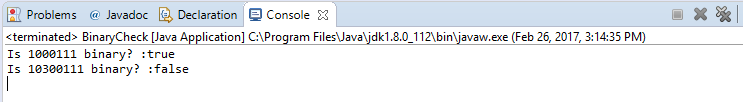
BinaryCheck mbc = **new** BinaryCheck();

System.***out***.println("Is 1000111 binary? :"+mbc.isBinaryNumber(1000111));

System.***out***.println("Is 10300111 binary? :"+mbc.isBinaryNumber(10300111));

}

}



1. Program: Write a program to generate 10 random numbers using Random class within the range of 1 to 100 using for loop.

Description:

By using Math.random() method also we can generate random number in java

randomNum = minimum + (int)(Math.random() \* maximum);

**package** CoreXJava;

**public** **class** Randomizer {

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

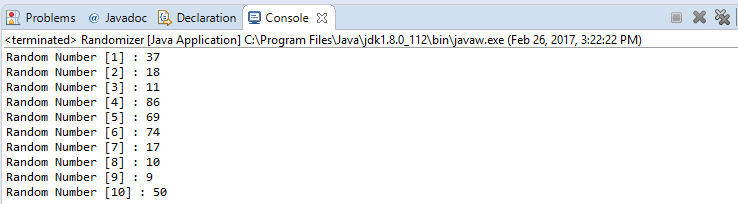
**for**(**int** i=0; i < 10 ; i++){

System.***out***.println("Random Number ["+ (i+1) + "] : " + (**int**)(Math.*random*()\*100));

}

}

}



1. Program: Write a program to use super() method.

Description:

Basically super keyword used to refer super class methods and variables.

**package** CoreXJava;

**class** Animal

{

Animal()

{

System.***out***.println("animal is created");

}

}

**class** Dog **extends** Animal

{

Dog()

{

**super**();

System.***out***.println("dog is created");

}

}

**class** TestSuper

{

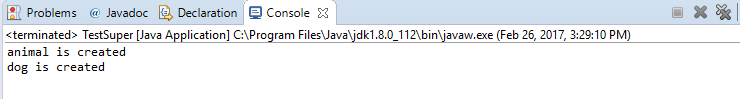
**public** **static** **void** main(String args[])

{

**new** Dog();

}

}



1. Program: Write a program to reverse vowels of a given string.

Description:

Let's string is InstanceOfJava

After reversing vowels in a string=anstancOefjavI

**package** CoreXJava;

**import** java.util.ArrayList;

**public** **class** ReverseVowel {

**public** String reverseVowels(String s) {

ArrayList<Character> vowList = **new** ArrayList<Character>();

vowList.add('a');

vowList.add('e');

vowList.add('i');

vowList.add('o');

vowList.add('u');

vowList.add('A');

vowList.add('E');

vowList.add('I');

vowList.add('O');

vowList.add('U');

**char**[] arr = s.toCharArray();

**int** i=0;

**int** j=s.length()-1;

**while**(i<j){

**if**(!vowList.contains(arr[i])){

i++;

**continue**;

}

**if**(!vowList.contains(arr[j])){

j--;

**continue**;

}

**char** t = arr[i];

arr[i]=arr[j];

arr[j]=t;

i++;

j--;

}

**return** **new** String(arr);

}

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

ReverseVowel rv = **new** ReverseVowel();

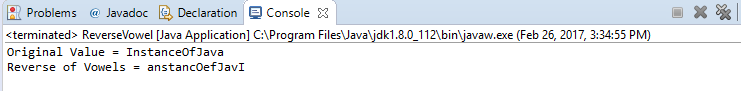
String str = "InstanceOfJava";

System.***out***.println("Original Value = "+str);

System.***out***.println("Reverse of Vowels = "+rv.reverseVowels(str));

}

}



17. WAP to implement Bubble sort in java.

**package** CoreXJava;

**import** java.util.Scanner;

**class** BubbleSort {

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

**int** n, c, d, swap;

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

System.***out***.println("Input number of integers to sort");

n = in.nextInt();

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

System.***out***.println("Enter " + n + " integers");

**for** (c = 0; c < n; c++)

array[c] = in.nextInt();

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

**for** (d = 0; d < n - c - 1; d++) {

**if** (array[d] > array[d+1]) /\* For descending order use < \*/

{

swap = array[d];

array[d] = array[d+1];

array[d+1] = swap;

}

}

}

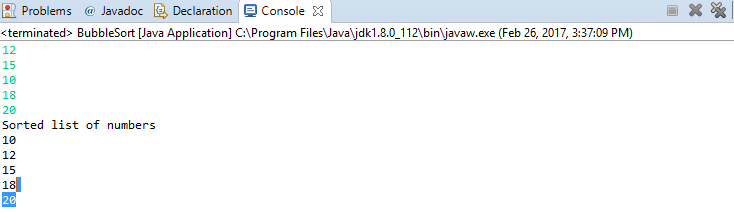
System.***out***.println("Sorted list of numbers");

**for** (c = 0; c < n; c++)

System.***out***.println(array[c]);

in.close();

}

}

18. WAP to print the following pattern (example3)

\*

\* \* \*

\* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \*

**package** CoreXJava;

**import** java.util.Scanner;

**class** Star

{

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

{

**int** n, c, d;

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

System.***out***.println("Enter the number of rows of star triangle you want");

n = in.nextInt();

System.***out***.println("Star triangle :-");

**for** ( c = 1 ; c <= n ; c++ )

{

**for** ( d = 1 ; d <= c ; d++ )

{

System.***out***.print("\* ");

}

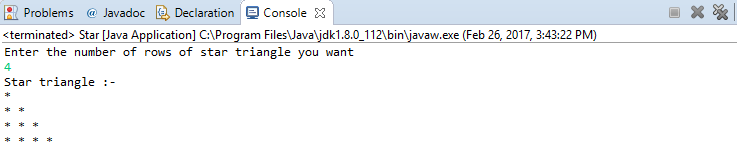
System.***out***.println();

in.close();

}

}

}



19. WAP to get factorial of a number without using recursion.

**package** CoreXJava;

**import** java.util.Scanner;

**public** **class** Factorial

{

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

{

**int** n, mul = 1;

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

System.***out***.print("Enter any integer:");

n = s.nextInt();

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

{

mul = mul \* i;

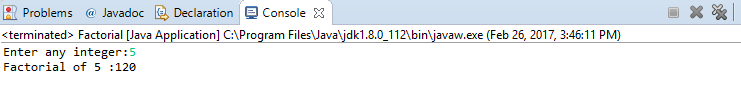
}

System.***out***.println("Factorial of "+n+" :"+mul);

s.close();

}

}



20. WAP to take a sentence from user and do unique word count using split().

**package** CoreXJava;

**public** **class** StringDemo

{

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

{

String str = "You can become Data Scientist After Big Data Developer";

String arr[] = str.split(" ");

System.***out***.println("Statement = "+str);

System.***out***.println("The number of words in the String are : "+arr.length);

}

}

