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.

Code:

```
package com.java2novice.algos;
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(22);
DuplicateNumber dn = new DuplicateNumber();
System.out.println("Duplicate Number:
"+dn.findDuplicateNumber(numbers));
}
}
```

Output:

```
Duplicate Number: 22
```

Next

PROGRAM: FIND OUT MIDDLE INDEX WHERE SUM OF BOTH ENDS ARE EQUAL.

Description:

You are given an array of numbers. Find out the array index or position where sum of numbers preceeding the index is equals to sum of numbers succeeding the index.

Code:

?

```
package com.java2novice.algos;

public class FindMiddleIndex {

    public static int findMiddleIndex(int[] numbers) throws Exception {

        int endIndex = numbers.length - 1;
        int startIndex = 0;
        int sumLeft = 0;
        int sumRight = 0;
        while (true) {
            if (sumLeft > sumRight) {
                sumRight += numbers[endIndex--];
            } else {
                sumLeft += numbers[startIndex++];
            }
            if (startIndex > endIndex) {
                if (sumLeft == sumRight) {
                    break;
                } else {
                    throw new Exception(
                            "Please pass proper array to match the requirement");
                }
            }
        }
        return endIndex;
    }

    public static void main(String a[]) {
        int[] num = { 2, 4, 4, 5, 4, 1 };
        try {
            System.out.println("Starting from index 0, adding numbers till index "
                            + findMiddleIndex(num) + " and");
            System.out.println("adding rest of the numbers can be equal");
        } catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
    }
}
```

Output:

```
Starting from index 0, adding numbers till index 2 and
adding rest of the numbers can be equal
```

- See more at: http://java2novice.com/java-interview-programs/find-middle-index/#sthash.X7dduUtm.dpuf

PROGRAM: WRITE A SINGLETON CLASS.

Description:

Singleton class means you can create only one object for the given class. You can create a singleton class by making its constructor as private, so that you can restrict the creation of the object. Provide a static method to get instance of the objec wherein you can handle the object creation inside the class only. In this example we are creating object by using static bloc

Code:

```
package com.java2novice.algos;

public class MySingleton {

		private static MySingleton myObj;

		static{
				myObj = new MySingleton();
		}

		private MySingleton(){

		}

		public static MySingleton getInstance(){
				return myObj;
		}

		public void testMe(){
				System.out.println("Hey.... it is working!!!");
		}

		public static void main(String a[]){
				MySingleton ms = getInstance();
				ms.testMe();
		}
}
```

- See more at: http://java2novice.com/java-interview-programs/java-singleton/#sthash.67HOhI4P.dpuf

PROGRAM: WRITE A PROGRAM TO REVERSE A STRING USING RECURSIVE ALGORITHM.

Description:

Write a program to reverse a string using recursive methods.
You should not use any string reverse methods to do this.

Code:

?

```
package com.java2novice.algos;

public class StringRecursiveReversal {

    String reverse = "";

    public String reverseString(String str){

        if(str.length() == 1){
            return str;
        } else {
            reverse += str.charAt(str.length()-1)
                    +reverseString(str.substring(0,str.length()-1));
            return reverse;
        }
    }

    public static void main(String a[]){
        StringRecursiveReversal srr = new StringRecursiveReversal();
        System.out.println("Result: "+srr.reverseString("Java2novice"));
    }
}
```

Output:

```
Result: ecivon2avaJ
```

- See more at: http://java2novice.com/java-interview-programs/string-reverse-recursive/#sthash.imiF87XE.dpuf

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 nume
operations.

Code:

?

```
package com.java2novice.algos;

public class NumberReverse {

    public int reverseNumber(int number){

        int reverse = 0;
        while(number != 0){
            reverse = (reverse*10)+(number%10);
            number = number/10;
        }
        return reverse;
    }

    public static void main(String a[]){
        NumberReverse nr = new NumberReverse();
        System.out.println("Result: "+nr.reverseNumber(17868));
    }
}
```

Output:

```
Result: 86871
```

- See more at: http://java2novice.com/java-interview-programs/reverse-number/#sthash.HPdhuJi2.dpuf

PROGRAM: WRITE A PROGRAM TO CONVERT DECIMAL NUMBER TO BINARY FORMAT.

Description:

Write a program to convert decimal number to binary format using numeric operations. Below example shows how to conv
decimal number to binary format using numeric operations.

Code:

?

```

package com.java2novice.algos;

public class DecimalToBinary {

    public void printBinaryFormat(int number){
        int binary[] = new int[25];
        int index = 0;
        while(number > 0){
            binary[index++] = number%2;
            number = number/2;
        }
        for(int i = index-1;i >= 0;i--){
            System.out.print(binary[i]);
        }
    }

    public static void main(String a[]){
        DecimalToBinary dtb = new DecimalToBinary();
        dtb.printBinaryFormat(25);
    }
}
```

```
21
```

Output:

```
11001
```

- See more at: http://java2novice.com/java-interview-programs/decimal-to-binary/#sthash.ZY6PHpkr.dpuf

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.
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:
(1 + 2 + 3 + 6) / 2 = 6.

Code:

?

```
package com.java2novice.algos;

public class IsPerfectNumber {

    public boolean isPerfectNumber(int number){

        int temp = 0;
        for(int i=1;i<=number/2;i++){
            if(number%i == 0){
                temp += i;
            }
        }
        if(temp == number){
            System.out.println("It is a perfect number");
            return true;
        } else {
            System.out.println("It is not a perfect number");
            return false;
        }
    }

    public static void main(String a[]){
        IsPerfectNumber ipn = new IsPerfectNumber();
        System.out.println("Is perfect number: "+ipn.isPerfectNumber(28));
    }
}
```

Output:

```
28
It is a perfect number
Is perfect number: true
```

- See more at: http://java2novice.com/java-interview-programs/perfect-number/#sthash.f1lvqJQO.dpuf

PROGRAM: WRITE A PROGRAM TO FIND MAXIMUM REPEATED WORDS FROM A FILE.

Description:

Write a program to read words from a file. Count the repeated or duplicated words. Sort it by maximum repeated or duplicated word count.

Code:

?

```java
package com.java2novice.algos;

import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Set;
import java.util.StringTokenizer;
import java.util.Map.Entry;

public class MaxDuplicateWordCount {

    public Map<String, Integer> getWordCount(String fileName){

        FileInputStream fis = null;
        DataInputStream dis = null;
        BufferedReader br = null;
        Map<String, Integer> wordMap = new HashMap<String, Integer>();
        try {
            fis = new FileInputStream(fileName);
            dis = new DataInputStream(fis);
            br = new BufferedReader(new InputStreamReader(dis));
            String line = null;
            while((line = br.readLine()) != null){
                StringTokenizer st = new StringTokenizer(line, " ");
                while(st.hasMoreTokens()){
                    String tmp = st.nextToken().toLowerCase();
                    if(wordMap.containsKey(tmp)){
                        wordMap.put(tmp, wordMap.get(tmp)+1);
                    } else {
                        wordMap.put(tmp, 1);
                    }
                }
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally{
            try{if(br != null) br.close();}catch(Exception ex){}
        }
        return wordMap;
    }

    public List<Entry<String, Integer>> sortByValue(Map<String, Integer> wordMap){

        Set<Entry<String, Integer>> set = wordMap.entrySet();
        List<Entry<String, Integer>> list = new ArrayList<Entry<String, Integer>>(
        Collections.sort( list, new Comparator<Map.Entry<String, Integer>>()
        {
            public int compare( Map.Entry<String, Integer> o1, Map.Entry<String, In
            {
                return (o2.getValue()).compareTo( o1.getValue() );
            }
        } );
        return list;
    }

    public static void main(String a[]){
        MaxDuplicateWordCount mdc = new MaxDuplicateWordCount();
        Map<String, Integer> wordMap = mdc.getWordCount("C:/MyTestFile.txt");
        List<Entry<String, Integer>> list = mdc.sortByValue(wordMap);
        for(Map.Entry<String, Integer> entry:list){
            System.out.println(entry.getKey()+" ==== "+entry.getValue());
        }
    }
}
```

Output:

```
one ==== 3
the ==== 3
```

```
that ==== 3
of ==== 2
in ==== 2
some ==== 2
to ==== 1
summary ==== 1
but ==== 1
have ==== 1
common ==== 1
least ==== 1
simplest ==== 1
```

- See more at: http://java2novice.com/java-interview-programs/max-repeated-words-file/#sthash.z6ESVMBN.dpuf

Or

```
public class findRepeatedWorkFromFile {
public static void main(String[] args) throws IOException {
File filehandle = new File("file.txt");
FileReader fileReader = new FileReader(filehandle);
BufferedReader bufferReader = new BufferedReader(fileReader)
HashMap<string, integer=""> hm = new HashMap<string, integer="">();
String line;
while ((line = bufferReader.readLine()) != null) {
String[] lineArray = line.split("");
for (String s : lineArray) {
if (hm.containsKey(s)) {
int count = hm.get(s);
count++;
hm.put(s, count);
} else {
hm.put(s, 1);
}
}
}
System.out.println(hm);
}
}
```

PROGRAM: WRITE A PROGRAM TO FIND OUT DUPLICATE CHARACTERS IN A STRING.

Description:
Write a program to find out duplicate or repeated characters in a string, and calculate the count of repeatation.

Code:
?

```
package com.java2novice.algos;
```

```
import java.util.HashMap;
import java.util.Map;
import java.util.Set;

public class DuplicateCharsInString {

    public void findDuplicateChars(String str){

        Map<Character, Integer> dupMap = new HashMap<Character, Integer>();
        char[] chrs = str.toCharArray();
        for(Character ch:chrs){
            if(dupMap.containsKey(ch)){
                dupMap.put(ch, dupMap.get(ch)+1);
            } else {
                dupMap.put(ch, 1);
            }
        }
        Set<Character> keys = dupMap.keySet();
        for(Character ch:keys){
            if(dupMap.get(ch) > 1){
                System.out.println(ch+"--->"+dupMap.get(ch));
            }
        }
    }

    public static void main(String a[]){
        DuplicateCharsInString dcs = new DuplicateCharsInString();
        dcs.findDuplicateChars("Java2Novice");
    }
}
```

Output:

```
v--->2
a--->2
```

- See more at: http://java2novice.com/java-interview-programs/duplicate-string-character-count/#sthash.Qi9JZeOw.dpuf

PROGRAM: WRITE A PROGRAM TO FIND TOP TWO MAXIMUM NUMBERS IN A ARRAY.

Description:

Write a program to find top two maximum numbers in the given 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.

Code:

?

```
package com.java2novice.algos;

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);
    }
}
```

Output:

```
First Max Number: 99
Second Max Number: 78
```

- See more at: http://java2novice.com/java-interview-programs/two-max-numbers-in-array/#sthash.kBJ3yhTi.dpuf

PROGRAM: WRITE A PROGRAM TO SORT A MAP BY VALUE.

Description:

Sort or order a HashMap or TreeSet or any map item by value. Write a comparator which compares by value, not by key. Entry class might hleps you here.

Code:

?

```
package com.java2novice.algos;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Set;
import java.util.Map.Entry;

public class OrderByValue {

    public static void main(String a[]){
        Map<String, Integer> map = new HashMap<String, Integer>();
        map.put("java", 20);
        map.put("C++", 45);
        map.put("Java2Novice", 2);
        map.put("Unix", 67);
        map.put("MAC", 26);
        map.put("Why this kolavari", 93);
        Set<Entry<String, Integer>> set = map.entrySet();
        List<Entry<String, Integer>> list = new ArrayList<Entry<String, Integer>>(
        Collections.sort( list, new Comparator<Map.Entry<String, Integer>>()
        {
            public int compare( Map.Entry<String, Integer> o1, Map.Entry<String, In
            {
                return (o2.getValue()).compareTo( o1.getValue() );
            }
        } );
        for(Map.Entry<String, Integer> entry:list){
            System.out.println(entry.getKey()+" ==== "+entry.getValue());
        }
    }
}
```

Output:

```
Why this kolavari ==== 93
Unix ==== 67
C++ ==== 45
MAC ==== 26
java ==== 20
Java2Novice ==== 2
```

- See more at: http://java2novice.com/java-interview-programs/sort-a-map-by-value/#sthash.QTOuHRuW.dpuf

PROGRAM: WRITE A PROGRAM TO FIND COMMON ELEMENTS BETWEEN TWO ARRAYS.

Description:

Write a program to identify common elements or numbers between two given arrays. You should not use any inbuilt methods are list to find common values.

Code:

?

```
package com.java2novice.algos;

public class CommonElementsInArray {

    public static void main(String a[]){
        int[] arr1 = {4,7,3,9,2};
        int[] arr2 = {3,2,12,9,40,32,4};
        for(int i=0;i<arr1.length;i++){
            for(int j=0;j<arr2.length;j++){
                if(arr1[i]==arr2[j]){
                    System.out.println(arr1[i]);
                }
            }
        }
    }
}
```

Output:

```
4
3
9
2
```

- See more at: http://java2novice.com/java-interview-programs/common-elements-in-two-arrays/#sthash.nvAexFhI.dpuf

PROGRAM: WRITE A PROGRAM TO PRINT FIBONACCI SERIES.

Description:

In mathematics, the Fibonacci numbers or Fibonacci series or Fibonacci sequence are the numbers in the following integer sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144... By definition, the first two numbers in the Fibonacci sequence are 0 ar and each subsequent number is the sum of the previous two. Below example shows how to create fibonacci series.

Code:

?

```
package com.java2novice.algos;

public class MyFibonacci {

    public static void main(String a[]){

         int febCount = 15;
         int[] feb = new int[febCount];
         feb[0] = 0;
```

```
            feb[1] = 1;
            for(int i=2; i < febCount; i++){
                feb[i] = feb[i-1] + feb[i-2];
            }

            for(int i=0; i< febCount; i++){
                    System.out.print(feb[i] + " ");
            }
        }
    }
```

Output:

```
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

- See more at: http://java2novice.com/java-interview-programs/fibonacci-series/#sthash.qWmGoLjr.dpuf

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 num is 259, then the sum should be 2+5+9 = 16.

Code:

?

```
package com.java2novice.algos;

public class MyNumberSumRec {

    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[]){
        MyNumberSumRec mns = new MyNumberSumRec();
        System.out.println("Sum is: "+mns.getNumberSum(223));
    }
}
```

Output:

```
Sum is: 7
```

- See more at: http://java2novice.com/java-interview-programs/number-sum-recursive/#sthash.TNKadDhE.dpuf

PROGRAM: WRITE A PROGRAM TO CHECK THE GIVEN NUMBER IS A PRIME NUMBER OR NOT?

Description:

A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself. A natura number greater than 1 that is not a prime number is called a composite number. For example, 5 is prime, as only 1 and 5 div it, whereas 6 is composite, since it has the divisors 2 and 3 in addition to 1 and 6. The fundamental theorem of arithmetic establishes the central role of primes in number theory: any integer greater than 1 can be expressed as a product of primes th is unique up to ordering. This theorem requires excluding 1 as a prime.

Code:

?

```java
package com.java2novice.algos;

public class MyPrimeNumCheck {

    public boolean isPrimeNumber(int number){

        for(int i=2; i<=number/2; i++){
            if(number % i == 0){
                return false;
            }
        }
        return true;
    }

    public static void main(String a[]){
        MyPrimeNumCheck mpc = new MyPrimeNumCheck();
        System.out.println("Is 17 prime number? "+mpc.isPrimeNumber(17));
        System.out.println("Is 19 prime number? "+mpc.isPrimeNumber(19));
        System.out.println("Is 15 prime number? "+mpc.isPrimeNumber(15));
    }
}
```

Output:

```
Is 17 prime number? true
Is 19 prime number? true
Is 15 prime number? false
```

- See more at: http://java2novice.com/java-interview-programs/is-prime-number/#sthash.1CGx6VeL.dpuf

PROGRAM: WRITE A PROGRAM TO FIND THE GIVEN NUMBER IS ARMSTRONG NUMBER OR NOT?

Description:
Armstrong numbers are the sum of their own digits to the power of the number of digits. It is also known as narcissistic numbers.

Code:
?

```
package com.java2novice.algos;

public class MyArmstrongNumber {

    public boolean isArmstrongNumber(int number){

        int tmp = number;
        int noOfDigits = String.valueOf(number).length();
        int sum = 0;
        int div = 0;
        while(tmp > 0)
        {
            div = tmp % 10;
            int temp = 1;
            for(int i=0;i<noOfDigits;i++){
                temp *= div;
            }
            sum += temp;
            tmp = tmp/10;
        }
        if(number == sum) {
            return true;
        } else {
            return false;
        }
    }

    public static void main(String a[]){
        MyArmstrongNumber man = new MyArmstrongNumber();
        System.out.println("Is 371 Armstrong number? "+man.isArmstrongNumber(371))
        System.out.println("Is 523 Armstrong number? "+man.isArmstrongNumber(523))
        System.out.println("Is 153 Armstrong number? "+man.isArmstrongNumber(153))
    }
}
```

Output:

```
Is 371 Armstrong number? true
Is 523 Armstrong number? false
Is 153 Armstrong number? true
```

- See more at: http://java2novice.com/java-interview-programs/armstrong-number/#sthash.Vp12g5R2.dpuf

PROGRAM: WRITE A PROGRAM TO CHECK THE GIVEN NUMBER IS BINARY NUMBER OR NOT?

Description:

The binary numeral system, or base-2 number system, represents numeric values using two symbols: 0 and 1. More specifically, the usual base-2 system is a positional notation with a radix of 2. Because of its straightforward implementatio digital electronic circuitry using logic gates, the binary system is used internally by almost all modern computers.

Code:

?

```


package com.java2novice.algos;

public class MyBinaryCheck {

    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[]){
        MyBinaryCheck mbc = new MyBinaryCheck();
        System.out.println("Is 1000111 binary? :"+mbc.isBinaryNumber(1000111));
        System.out.println("Is 10300111 binary? :"+mbc.isBinaryNumber(10300111));
    }
}


```

Output:

```
Is 1000111 binary? :true
Is 10300111 binary? :false
```

- See more at: http://java2novice.com/java-interview-programs/is-binary-number/#sthash.37ifJLbb.dpuf

PROGRAM: WRITE A PROGRAM FOR BUBBLE SORT IN JAVA.

Description:

Bubble sort is a simple sorting algorithm that works by repeatedly stepping through the list to be sorted, comparing each pa adjacent items and swapping them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. The algorithm gets its name from the way smaller elements bubble to the top the list. Because it only uses comparisons to operate on elements, it is a comparison sort. You can see the code implementat below:

Code:

?

```java
package com.java2novice.algos;

public class MyBubbleSort {

    // logic to sort the elements
    public static void bubble_srt(int array[]) {
        int n = array.length;
        int k;
        for (int m = n; m >= 0; m--) {
            for (int i = 0; i < n - 1; i++) {
                k = i + 1;
                if (array[i] > array[k]) {
                    swapNumbers(i, k, array);
                }
            }
            printNumbers(array);
        }
    }

    private static void swapNumbers(int i, int j, int[] array) {

        int temp;
        temp = array[i];
        array[i] = array[j];
        array[j] = temp;
    }

    private static void printNumbers(int[] input) {

        for (int i = 0; i < input.length; i++) {
            System.out.print(input[i] + ", ");
        }
        System.out.println("\n");
    }

    public static void main(String[] args) {
        int[] input = { 4, 2, 9, 6, 23, 12, 34, 0, 1 };
        bubble_srt(input);

    }
}
```

Output:

```
2, 4, 6, 9, 12, 23, 0, 1, 34,

2, 4, 6, 9, 12, 0, 1, 23, 34,

2, 4, 6, 9, 0, 1, 12, 23, 34,

2, 4, 6, 0, 1, 9, 12, 23, 34,

2, 4, 0, 1, 6, 9, 12, 23, 34,

2, 0, 1, 4, 6, 9, 12, 23, 34,

0, 1, 2, 4, 6, 9, 12, 23, 34,

0, 1, 2, 4, 6, 9, 12, 23, 34,

0, 1, 2, 4, 6, 9, 12, 23, 34,

0, 1, 2, 4, 6, 9, 12, 23, 34,
```

- See more at: http://java2novice.com/java-interview-programs/bubble-sort/#sthash.hKDHSY4o.dpuf

PROGRAM: WRITE A PROGRAM FOR INSERTION SORT IN JAVA.

Description:

Insertion sort is a simple sorting algorithm that builds the final sorted array one item at a time. It is much less efficient on la lists than more advanced algorithms such as quicksort, heapsort, or merge sort. Every repetition of insertion sort removes ar element from the input data, inserting it into the correct position in the already-sorted list, until no input elements remain. T choice of which element to remove from the input is arbitrary, and can be made using almost any choice algorithm. You car the code implementation below:

Code:

?

```
package com.java2novice.algos;

public class MyInsertionSort {

    public static void main(String[] args) {

        int[] input = { 4, 2, 9, 6, 23, 12, 34, 0, 1 };
        insertionSort(input);
    }

    private static void printNumbers(int[] input) {

        for (int i = 0; i < input.length; i++) {
            System.out.print(input[i] + ", ");
        }
        System.out.println("\n");
    }

    public static void insertionSort(int array[]) {
        int n = array.length;
        for (int j = 1; j < n; j++) {
            int key = array[j];
            int i = j-1;
            while ( (i > -1) && ( array [i] > key ) ) {
                array [i+1] = array [i];
                i--;
            }
            array[i+1] = key;
            printNumbers(array);
        }
    }
}
```

Output:

```
2, 4, 9, 6, 23, 12, 34, 0, 1,

2, 4, 9, 6, 23, 12, 34, 0, 1,

2, 4, 6, 9, 23, 12, 34, 0, 1,

2, 4, 6, 9, 23, 12, 34, 0, 1,

2, 4, 6, 9, 12, 23, 34, 0, 1,

2, 4, 6, 9, 12, 23, 34, 0, 1,

0, 2, 4, 6, 9, 12, 23, 34, 1,

0, 1, 2, 4, 6, 9, 12, 23, 34,
```

- See more at: http://java2novice.com/java-interview-programs/insertion-sort/#sthash.3ZAsFNtU.dpuf

PROGRAM: WRITE A PROGRAM TO IMPLEMENT HASHCODE AND EQUALS.

Description:

The hashcode of a Java Object is simply a number, it is 32-bit signed int, that allows an object to be managed by a hash-bas data structure. We know that hash code is an unique id number allocated to an object by JVM. But actually speaking, Hash code is not an unique number for an object. If two objects are equals then these two objects should return same hash code. S we have to implement hashcode() method of a class in such way that if two objects are equals, ie compared by equal() method of that class, then those two objects must return same hash code. If you are overriding hashCode you need to override equals method also.

The below example shows how to override equals and hashcode methods. The class Price overrides equals and hashcode. If notice the hashcode implementation, it always generates unique hashcode for each object based on their state, ie if the objec state is same, then you will get same hashcode. A HashMap is used in the example to store Price objects as keys. It shows though we generate different objects, but if state is same, still we can use this as key.

Code:

?

```
package com.java2novice.algos;

import java.util.HashMap;

public class MyHashcodeImpl {

    public static void main(String a[]){

        HashMap<Price, String> hm = new HashMap<Price, String>();
        hm.put(new Price("Banana", 20), "Banana");
        hm.put(new Price("Apple", 40), "Apple");
        hm.put(new Price("Orange", 30), "Orange");
        //creating new object to use as key to get value
        Price key = new Price("Banana", 20);
        System.out.println("Hashcode of the key: "+key.hashCode());
        System.out.println("Value from map: "+hm.get(key));
    }
}

class Price{

    private String item;
    private int price;

    public Price(String itm, int pr){
        this.item = itm;
        this.price = pr;
    }

    public int hashCode(){
        System.out.println("In hashcode");
        int hashcode = 0;
        hashcode = price*20;
        hashcode += item.hashCode();
        return hashcode;
    }
```

```
      public boolean equals(Object obj){
          System.out.println("In equals");
          if (obj instanceof Price) {
              Price pp = (Price) obj;
              return (pp.item.equals(this.item) && pp.price == this.price);
          } else {
              return false;
          }
      }

      public String getItem() {
          return item;
      }
      public void setItem(String item) {
          this.item = item;
      }
      public int getPrice() {
          return price;
      }
      public void setPrice(int price) {
          this.price = price;
      }

      public String toString(){
          return "item: "+item+"  price: "+price;
      }
  }
```

Output:

```
In hashcode
In hashcode
In hashcode
In hashcode
Hashcode of the key: 1982479637
In hashcode
In equals
Value from map: Banana
```

- See more at: http://java2novice.com/java-interview-programs/equals-hashcode/#sthash.eDRdRDHS.dpuf

PROGRAM: HOW TO GET DISTINCT ELEMENTS FROM AN ARRAY BY AVOIDING DUPLICATE ELEMENTS?

Description:

The below example shows how to avoid duplicate elements from an array and disply only distinct elements. Please use only arrays to process it.

Code:

?

```
package com.java2novice.algos;

public class MyDisticntElements {

    public static void printDistinctElements(int[] arr){

        for(int i=0;i<arr.length;i++){
            boolean isDistinct = false;
            for(int j=0;j<i;j++){
                if(arr[i] == arr[j]){
                    isDistinct = true;
                    break;
                }
            }
            if(!isDistinct){
                System.out.print(arr[i]+" ");
            }
        }
    }

    public static void main(String a[]){

        int[] nums = {5,2,7,2,4,7,8,2,3};
        MyDisticntElements.printDistinctElements(nums);
    }
}
```

Output:

```
5 2 7 4 8 3
```

- See more at: http://java2novice.com/java-interview-programs/distinct-elements/#sthash.zUK0oJaq.dpuf

PROGRAM: WRITE A PROGRAM TO GET DISTINCT WORD LIST FROM THE GIVEN FILE.

Description:

Write a program to find all distinct words from the given file. Remove special chars like ".,;:" etc. Ignore case sensitivity.

Code:

?

```java
package com.java2novice.algos;

import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;
import java.util.StringTokenizer;

public class MyDistinctFileWords {

    public List<String> getDistinctWordList(String fileName){

        FileInputStream fis = null;
        DataInputStream dis = null;
        BufferedReader br = null;
        List<String> wordList = new ArrayList<String>();
        try {
            fis = new FileInputStream(fileName);
            dis = new DataInputStream(fis);
            br = new BufferedReader(new InputStreamReader(dis));
            String line = null;
            while((line = br.readLine()) != null){
                StringTokenizer st = new StringTokenizer(line, " ,.;:\"");
                while(st.hasMoreTokens()){
                    String tmp = st.nextToken().toLowerCase();
                    if(!wordList.contains(tmp)){
                        wordList.add(tmp);
                    }
                }
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally{
            try{if(br != null) br.close();}catch(Exception ex){}
        }
        return wordList;
    }

    public static void main(String a[]){

        MyDistinctFileWords distFw = new MyDistinctFileWords();
        List<String> wordList = distFw.getDistinctWordList("C:/sample.txt");
        for(String str:wordList){
            System.out.println(str);
        }
    }
}
```

Output:

```
the
while
statement
verifies
condition
before
entering
into
loop
to
see
whether
next
iteration
should
occur
or
not
do-while
executes
first
without
checking
it
after
finishing
each
will
always
execute
body
of
a
at
least
once
```

- See more at: http://java2novice.com/java-interview-programs/distinct-word-list/#sthash.4sp2JOXK.dpuf

PROGRAM: WRITE A PROGRAM TO GET A LINE WITH MAX WORD COUNT FROM THE GIVEN FILE.

Description:

Below example shows how to find out the line with maximum number of word count in the given file. In case if it has multi
lines with max number of words, then it has to list all those lines.

Code:

?

```java
package com.java2novice.algos;

import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;

public class MaxWordCountInLine {

    private int currentMaxCount = 0;
    private List<String> lines = new ArrayList<String>();

    public void readMaxLineCount(String fileName){

        FileInputStream fis = null;
        DataInputStream dis = null;
        BufferedReader br = null;

        try {
            fis = new FileInputStream(fileName);
            dis = new DataInputStream(fis);
            br = new BufferedReader(new InputStreamReader(dis));
            String line = null;
            while((line = br.readLine()) != null){

                int count = (line.split("\\s+")).length;
                if(count > currentMaxCount){
                    lines.clear();
                    lines.add(line);
                    currentMaxCount = count;
                } else if(count == currentMaxCount){
                    lines.add(line);
                }
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally{
            try{
                if(br != null) br.close();
            }catch(Exception ex){}
        }
    }

    public int getCurrentMaxCount() {
        return currentMaxCount;
    }

    public void setCurrentMaxCount(int currentMaxCount) {
        this.currentMaxCount = currentMaxCount;
    }

    public List<String> getLines() {
        return lines;
    }

    public void setLines(List<String> lines) {
        this.lines = lines;
    }

    public static void main(String a[]){

        MaxWordCountInLine mdc = new MaxWordCountInLine();
        mdc.readMaxLineCount("/Users/ngootooru/MyTestFile.txt");
        System.out.println("Max number of words in a line is: "+mdc.getCurrentMaxC
        System.out.println("Line with max word count:");
        List<String> lines = mdc.getLines();
        for(String l:lines){
            System.out.println(l);
        }
    }
}
```

MyTestFile.txt:

```
true, false, and null might seem like keywords, but they are actually literals.
You cannot use them as identifiers in your programs. The servlet context
is an interface which helps to communicate with other servlets. It contains
information about the Web application and container. It is kind of
application environment. Using the context, a servlet can obtain URL
references to resources, and store attributes that other servlets in the
context can use.
```

Output:

```
Max number of words in a line is: 13
Line with max word count:
true, false, and null might seem like keywords, but they are actually literals.
```

- See more at: http://java2novice.com/java-interview-programs/line-max-word-count/#sthash.BOBzOpgL.dpuf

PROGRAM: WRITE A PROGRAM TO CONVERT STRING TO NUMBER WITHOUT USING INTEGER.PARSEINT() METHOD.

Description:
Below example shows how to convert string format of a number to number without calling Integer.parseInt() method. We c
do this by converting each character into ascii format and form the number.

Code:
?

```
package com.java2novice.algos;

public class MyStringToNumber {

    public static int convert_String_To_Number(String numStr){

        char ch[] = numStr.toCharArray();
        int sum = 0;
        //get ascii value for zero
        int zeroAscii = (int)'0';
        for(char c:ch){
            int tmpAscii = (int)c;
            sum = (sum*10)+(tmpAscii-zeroAscii);
        }
        return sum;
    }

    public static void main(String a[]){

        System.out.println("\"3256\" == "+convert_String_To_Number("3256"));
        System.out.println("\"76289\" == "+convert_String_To_Number("76289"));
        System.out.println("\"90087\" == "+convert_String_To_Number("90087"));
    }
}
```

Output:

```
"3256" == 3256
"76289" == 76289
"90087" == 90087
```

- See more at: http://java2novice.com/java-interview-programs/string-to-number/#sthash.LNvGkmdQ.dpuf

PROGRAM: WRITE A PROGRAM TO FIND TWO LINES WITH MAX CHARACTERS IN DESCENDING ORDER.

Description:

Write a program to read a multiple line text file and write the 'N' longest lines to the output console, where the file to be rea
specified as command line aruguments. The program should read an input file. The first line should contain the value of the
number 'N' followed by multiple lines. 'N' should be a valid positive integer.

Code:

?

```
package com.longest.lines;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.Comparator;
import java.util.Set;
import java.util.TreeSet;

public class Main {

    public static void main(String[] args) {

        BufferedReader br = null;
        String filePath = args[0];
        int topList = 0;
        Set<Entries> liSet = new TreeSet<Entries>(new MyComp());
        try {
            br = new BufferedReader(new FileReader(new File(filePath)));
            String line = br.readLine();
            topList = Integer.parseInt(line.trim());
            while((line = br.readLine()) != null){
                line = line.trim();
                if(!"".equals(line)){
                    liSet.add(new Entries(line.length(), line));
                }
            }
            int count = 0;
            for(Entries ent:liSet){
                System.out.println(ent.line);
                if(++count == topList){
                    break;
                }
            }
        } catch (FileNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    public static class Entries{
        Integer length;
        String line;
        public Entries(Integer l,String line){
            length = l;
            this.line = line;
        }
    }

    public static class MyComp implements Comparator<Entries>{

        @Override
        public int compare(Entries e1, Entries e2) {
            if(e2.length > e1.length){
                return 1;
            } else {
                return -1;
            }
        }

    }
}
```

Sample input file:

```
3
Java2novice
My Test line 123

Java world
I know java language

This is a test program
java is simple
```

Output:

```
This is a test program
I know java language
My Test line 123
```

- See more at: http://java2novice.com/java-interview-programs/line-word-desc-order/#sthash.pf8iZRv1.dpuf

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.

Code:
?

```
package com.primesum;

public class Main {

    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;
    }
}
```

Output:

```
3682913
```

PROGRAM: FIND LONGEST SUBSTRING WITHOUT REPEATING CHARACTERS.

Description:

Given a string, find the longest substrings without repeating characters. Iterate through the given string, find the longest maximum substrings.

Code:

?

```
package com.java2novice.algos;

import java.util.HashSet;
import java.util.Set;

public class MyLongestSubstr {

    private Set<String> subStrList = new HashSet<String>();
    private int finalSubStrSize = 0;

    public Set<String> getLongestSubstr(String input){
        //reset instance variables
        subStrList.clear();
        finalSubStrSize = 0;
        // have a boolean flag on each character ascii value
        boolean[] flag = new boolean[256];
        int j = 0;
        char[] inputCharArr = input.toCharArray();
        for (int i = 0; i < inputCharArr.length; i++) {
            char c = inputCharArr[i];
            if (flag[c]) {
                extractSubString(inputCharArr,j,i);
                for (int k = j; k < i; k++) {
                    if (inputCharArr[k] == c) {
                        j = k + 1;
                        break;
                    }
                    flag[inputCharArr[k]] = false;
                }
            } else {
                flag[c] = true;
            }
        }
        extractSubString(inputCharArr,j,inputCharArr.length);
        return subStrList;
    }

    private String extractSubString(char[] inputArr, int start, int end){

        StringBuilder sb = new StringBuilder();
        for(int i=start;i<end;i++){
            sb.append(inputArr[i]);
        }
        String subStr = sb.toString();
        if(subStr.length() > finalSubStrSize){
            finalSubStrSize = subStr.length();
            subStrList.clear();
            subStrList.add(subStr);
        } else if(subStr.length() == finalSubStrSize){
            subStrList.add(subStr);
        }

        return sb.toString();
    }

    public static void main(String a[]){
        MyLongestSubstr mls = new MyLongestSubstr();
        System.out.println(mls.getLongestSubstr("java2novice"));
        System.out.println(mls.getLongestSubstr("java_language_is_sweet"));
        System.out.println(mls.getLongestSubstr("java_java_java_java"));
        System.out.println(mls.getLongestSubstr("abcabcbb"));
    }
}
```

Output:

```
[a2novice]
[uage_is]
[_jav, va_j]
[cab, abc, bca]
```

- See more at: http://java2novice.com/java-interview-programs/longest-substring/#sthash.IGNuWUp5.dpuf

- See more at: http://java2novice.com/java-interview-programs/prime-sum/#sthash.S5wmt8gj.dpuf

PROGRAM: WRITE A PROGRAM TO REMOVE DUPLICATES FROM SORTED ARRAY.

Description:

Given array is already sorted, and it has duplicate elements. Write a program to remove duplicate elements and return new
without any duplicate elements. The array should contain only unique elements.

Code:

?

```
package com.java2novice.algos;
```

```java
public class MyDuplicateElements {

    public static int[] removeDuplicates(int[] input){

        int j = 0;
        int i = 1;
        //return if the array length is less than 2
        if(input.length < 2){
            return input;
        }
        while(i < input.length){
            if(input[i] == input[j]){
                i++;
            }else{
                input[++j] = input[i++];
            }
        }
        int[] output = new int[j+1];
        for(int k=0; k<output.length; k++){
            output[k] = input[k];
        }

        return output;
    }

    public static void main(String a[]){
        int[] input1 = {2,3,6,6,8,9,10,10,10,12,12};
        int[] output = removeDuplicates(input1);
        for(int i:output){
            System.out.print(i+" ");
        }
    }
}
```

Output:

```
2 3 6 8 9 10 12
```

- See more at: http://java2novice.com/java-interview-programs/remove-duplicates-sorted-array/#sthash.p2QtzOoC.dpuf

Reverse a string without using string function in java

Posted by: instance of java Posted date: **18:22** / comment : 0

```
package com.instaceofjava;

public class ReverseString {

public static void main(String[] args) {

String str="Hello world";
String revstring="";

for(int i=str.length()-1;i>=0;--i){
revstring +=str.charAt(i);
}

System.out.println(revstring);
}
}
```

Program to print prime numbers in java

Posted by: instance of java Posted date: **18:18** / comment : 0

```
package com.instaceofjava;

public class primenumbers {

public static void main(String[] args) {

// TODO Auto-generated method stub

int num=50;

int count=0;

for(int i=1;i<=num;i++){

count=0;

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

if(i%j==0){

count++;
break;
}

}

if(count==0){

System.out.println(i);
}

}

}

}
```

```
Output:
1
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
```

Sorting string without using string Methods?

Posted by: instance of java Posted date: **18:46** / comment : 0

```
package com.Instanceofjava;
public class SortString {

public static void main(String[] args) {

String original = "edcba";
int j=0;
char temp=0;

 char[] chars = original.toCharArray();

 for (int i = 0; i <chars.length; i++) {

     for ( j = 0; j < chars.length; j++) {

      if(chars[j]>chars[i]){
           temp=chars[i];
```

```
          chars[i]=chars[j];
          chars[j]=temp;
      }

  }

}

for(int k=0;k<chars.length;k++){
System.out.println(chars[k]);
}

}

}
```

```
```

Output:

```
abcde
```

Sort the string using String Methods?

Posted by: instance of java Posted date: **18:44** / comment : 0

```
package com.instanceofjava;

public class SortString {

public static void main(String[] args) {

   String original = "edcba";

  char[] chars = original.toCharArray();

 Arrays.sort(chars);

 String sorted = new String(chars);
 System.out.println(sorted);
}

}
```

OutPut:

```
abcde
```

Fibonacci series without using recursion in java

Posted by: instance of java Posted date: **18:34** / comment : 0

```
package com.instaceofjava;

public class Fibanaciwithoutrecursive {

public static void main(String[] args) {

int n1=0;

int n2=1;

System.out.println(n1);

System.out.println(n2);

for(int i=0;i<=100;i++){

int sum=n1+n2;

if(sum<=100){

n1=n2;

n2=sum;

System.out.println(sum);

}
}
}
```

```
Output:
0
1
1
2
3
5
8
13
21
34
55
89
```

Fibonacci series using recursion in java

Posted by: instance of java Posted date: **18:40** / comment : 0

```
package com.instaceofjavaforus;
 public class FibanacciRecursive {
public void fibanci(int n1,int n2){
 int sum=0;
 if(n1==0){
 System.out.println(n1+"\n"+n2);
 }
 sum=n1+n2;
 if(sum<=100){
  System.out.println(sum);
  n1=n2;
  n2=sum;

  fibanci(n1,n2);

 }
}

public static void main(String[] args) {

     FibanacciRecursive fb=new FibanacciRecursive();
     fb.fibanci(0,1);
}

}
```

```
Output:

0

1

1

2

3

5

8

13

21

34
```

```
55

89
```

How to find largest element in an array with index and value using array?

Posted by: instance of java Posted date: **18:58** / comment : 0

```
package com.instanceofjava;
public class Array {

public static void main(String[] args) {

int arr[]={1,120,56,78,87};
int largest=arr[0];
int smallest=arr[0];
int small=0;
int index=0;

for(int i=1;i<arr.length;i++){

if(arr[i]>largest){

largest=arr[i];
index=i;

}
else if(largest>arr[i]){

smallest=arr[i];
small=i;

}
}

System.out.println(largest);
System.out.println(index);
System.out.println(smallest);
System.out.println(small);

}

}
```

Output:

```
120
1
87
```

```
4
```

Sort integer array using Bubble Sort in java

Posted by: instance of java Posted date: **19:02** / comment : 0

```
package com.instaceofjava;

public class Bubblesort {

public static void main(String[] args) {

int a[]={23,5,6,66,1};

int n=a.length;

       int temp=0;

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

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

if(a[j-1]>a[j]){
   temp=a[j];
  a[j]=a[j-1];
  a[j-1]=temp;

  }

}

}

for(int k=0;k<n;k++){
System.out.println(a[k]);
}

}

}
```

Output:

```
1,
5
6
23
```

```
66
```

Object cloning in java example

Posted by: instance of java Posted date: **19:07** / comment : 0

```
package com.instanceofjava;

public class Employee implements Cloneable {

        int a=0;
        String name="";
        Employee (int a,String name){
        this.a=a;
        this.name=name;
}

public Employee clone() throws CloneNotSupportedException{

return (Employee ) super.clone();

}

public static void main(String[] args) {

          Employee e=new Employee (2,"Indhu");
            System.out.println(e.name);

try {

 Employee b=e.clone();
 System.out.println(b.name);

}
 catch (CloneNotSupportedException e1) {

e1.printStackTrace();
}
}

}
```

Output:

```
Indhu
Indhu
```

add two matrices

Posted by: instance of java Posted date: **20:48** / comment : 0

```java
package com.instanceofjava;

import java.util.Scanner;

class Add2Matrix
{

   public static void main(String args[])
   {

      int rows, cols, c, d;

      Scanner in = new Scanner(System.in);

      System.out.println("Please Enter number of rows and columns");

      rows = in.nextInt();
      cols  = in.nextInt();

      int first[][] = new int[rows][cols];
      int second[][] = new int[rows][cols];
      int sum[][] = new int[rows][cols];

      System.out.println("Please Enter elements of first matrix");

      for (  c = 0 ; c < rows ; c++ )
         for ( d = 0 ; d < cols ; d++ )
            first[c][d] = in.nextInt();

      System.out.println("Please Enter elements of second matrix");

      for ( c = 0 ; c < rows ; c++ )
         for ( d = 0 ; d < cols ; d++ )
            second[c][d] = in.nextInt();

      for ( c = 0 ; c < rows ; c++ )
         for ( d = 0 ; d < cols ; d++ )
             sum[c][d] = first[c][d] + second[c][d];  //replace '+' with '-' to subtract matrices

      System.out.println("Sum of entered matrices:-");

      for ( c = 0 ; c < rows ; c++ )
      {
         for ( d = 0 ; d < cols ; d++ )
            System.out.print(sum[c][d]+"\t");
         System.out.println();
      }
   }

}
```

```

```

Output:

```
Please Enter number of rows and columns

3
3

Please Enter elements of first matrix

1 1 1
1 1 1
1 1 1

Please Enter elements of second matrix

2 2 2
2 2 2
2 2 2

Sum of entered matrices:-

3   3   3
3   3   3
3   3   3
```

Print 1 to 10 without using loop in java?

Posted by: instance of java Posted date: **15:01** / comment : 0

- Basically to display numbers from 1 to 10 or a series we will use for , while or do while loop
- So here is the programs to do same thing without using loop.
- This is possible in two ways
- First one is to display by printing all those things using system.out.println.
- second one is by using recursive method call lets see these two programs

Solution #1:

```
package com.instanceofjavaTutorial;

class Demo{

public static void main(String args[]) {

    System.out.println(1);
    System.out.println(2);
    System.out.println(3);
    System.out.println(4);
    System.out.println(5);
    System.out.println(6);
    System.out.println(7);
    System.out.println(8);
    System.out.println(9);
    System.out.println(10);

}

}
```

Output:

```
1
2
 3
 4
5
6
7
8
9
10
```

Solution #2

```
package com.instanceofjavaTutorial;
class PrintDemo{

public static void recursivefun(int n)
{

  if(n <= 10) {

       System.out.println(n);
         recursivefun(n+1);   }
}

public static void main(String args[])
{

recursivefun(1);

 }

}
```

Output:

```
1
2
 3
 4
5
6
7
8
9
10
```

Convert Byte Array to String

Posted by: instance of java Posted date: **09:01** / comment : 2

- To convert byte array to string we have a constructor in String class which is taking byte array as an argument.
- So just we need to pass byte array object to string as argument while creating String class object.

```
 package com.instanceofjavaforus;


public class ByteArrayToString {
   /*
   * This method converts a byte array to a String object.
     */

    public static void convertByteArrayToString() {

        byte[] byteArray = new byte[] {78,73, 67,69};
        String value = new String(byteArray);
       System.out.println(value);

    }

    public static void main(String[] args) {
        ByteArrayToString.convertByteArrayToString();
    }

}
```

Output:

```
NICE
```

Convert String to Byte Array:

```
 package com.instanceofjava;

public class StringTOByteArray{
   /*
   * This example shows how to convert a String object to byte array.
     */

    public static void main(String[] args) {

    String data = "Instance of java";

    byte[] byteData = data.getBytes();

    System.out.println(byteData);

    }

}
```

Output:

```
[B@3b26456a
```

Remove duplicates from an array java

Posted by: instance of java Posted date: **11:09** / comment : 0

Removing duplicate elements form an array using collections:

```
package com.instanceofjavaTutorial;

import java.util.Arrays;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

public class RemoveDupArray {

  public static void main(String[] args) {

  // A string array with duplicate values
   String[] data = { "E", "C", "B", "E", "A", "B", "E", "D", "B", "A" };

System.out.println("Original array         : " + Arrays.toString(data));

 List<String> list = Arrays.asList(data);
 Set<String> set = new HashSet<String>(list);

  System.out.print("After removing duplicates: ");
  String[] resarray= new String[set.size()];
   set.toArray(resarray);

 for (String ele: resarray) {

 System.out.print(ele + ", ");

 }

}

}
```

```
OutPut:
Original array         : [E, C, B, E, A, B, E, D, B, A]
After removing duplicates: D, E, A, B, C
```

Program Check even or odd without using modulus and division operators

Posted by: instance of java Posted date: **14:49** / comment : 3

- Checking the number even or odd program is very easy. Anybody can solve this but there is a condition we need to see.
- When we get this question in interview "write a program to check given number is even or odd" always continues with "without using modulus and division operators".
- Before going to actual program lets see how to check a number is even or odd by using modulus and division operators.

Program to check number is even or odd by using modulus "%" operator

```
package instanceofjava;
import java.util.Scanner;

public class EvenorOdd {

public static void main(String []args )    {

    int number;
    Scanner in= new Scanner(System.in);

    System.out.println("Enter a number to check even or odd");
    number=in.nextInt();

    if((number % 2)==0){
        System.out.println(+number+" is Even number");
    }else{
        System.out.println(+number+" is Odd Number");
    }

}
}
```

Output:

```
Enter a number to check even or odd
37
37 is Odd Number
```

How to Add elements to hash map and Display?

Posted by: instance of java Posted date: **18:54** / comment : 0

```
package com.instaceofjava;

import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;

public class Mapiterator {

public static void main(String[] args) {

HashMap map=new HashMap();

map.put(1, "indhu");
map.put("d", "sindhu");
map.put("3", "swathi");

if(!map.isEmpty()){

Iterator it=map.entrySet().iterator();

while(it.hasNext()){

Map.Entry obj=(Entry) it.next();
System.out.println(obj.getValue());

}

}

}

}
```

Output:

```
swathi
indhu
sindhu
```

Sort ArrayList in descending order

Posted by: instance of java Posted date: **10:49** / comment : 0

Descending order:

```
package com.instanceofjavaforus;
import java.util.ArrayList;
 import java.util.Collections;
import java.util.Comparator;

public class SortArrayListDesc {

     public static void main(String[] args) {

            //create an ArrayList object
            ArrayList arrayList = new ArrayList();

            //Add elements to Arraylist
            arrayList.add(1);
            arrayList.add(2);
            arrayList.add(3);
           arrayList.add(4);
            arrayList.add(5);
            arrayList.add(6);

            /*
           Use static Comparator reverseOrder() method of Collections
        utility class to get comparator object
           */

         Comparator comparator = Collections.reverseOrder();

          System.out.println("Before sorting  : "  + arrayList);


           /*
              use
```

```
             static void sort(List list, Comparator com) method of Collections class.
           */

           Collections.sort(arrayList,comparator);
           System.out.println("After sorting  : + arrayList);


     }
  }
```

```
OutPut:
Before sorting  : [1, 2, 3, 4, 5, 6]
After sorting  : [6, 5, 4, 3, 2, 1]
```

Ascending order:

```
package com.instanceofjavaforus;
import java.util.ArrayList;
 import java.util.Collections;
import java.util.Comparator;

public class SortArrayListAsc{

     public static void main(String[] args) {

           //create an ArrayList object
           ArrayList arrayList = new ArrayList();

           //Add elements to Arraylist
           arrayList.add(10);
           arrayList.add(4);
           arrayList.add(7);
          arrayList.add(2);
           arrayList.add(5);
           arrayList.add(3);



         System.out.println("Before sorting  : "  + arrayList);


          /*
             use
             static void sort(List list) method of Collections class.
           */

```

```
            Collections.sort(arrayList);
            System.out.println("After sorting  : + arrayList);


     }
   }
```

```
OutPut:
Before sorting  : [10, 4, 7, 2, 5, 3]
After sorting  : [2, 3, 4, 5, 7, 10]
```

Sort object using comparator

Posted by: instance of java Posted date: **19:18** / comment : 0

```
package com.instanceofjava;

public class Employee
{

 int id;
 String name;

 public Employee(int id, String name) {

  this.id=id;
  this.name=name;

 }

public int getId() {

return id;

}

 public void setId(int id) {
  this.id = id;
 }

 public String getName() {
  return name;
 }

public void setName(String name) {

  this.name = name;

}
```

```java

}
```

```java
class MyEmpComp implements Comparator<Employee >
{

 @Override
 public int compare(Employee e1, Employee e2) {

        if(e1.getName() < e2.getName()){
            return 1;
        } else {
            return -1;
        }

}

}
```

```java
package com.oops;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;

import java.io.*;

class Main{

public static void main(String args[]){

ArrayList al=new ArrayList();

al.add(new Employee(101,"Indhu"));
al.add(new Employee(106,"Sindhu"));
al.add(new Employee(105,"Swathi"));

Collections.sort(al,MyEmpComp );

Iterator itr=al.iterator();

while(itr.hasNext()){

Employee st=(Employee)itr.next();
System.out.println(st.id +" "+st.name );

 }


}


}
```

Output:

```
Indhu
Swathi
Sindhu
```

Count the number of occurrences of a character in a String?

Posted by: instance of java Posted date: **18:52** / comment : 0

```
package com.instaceofjava;

public class StringCount {

public static void main(String[] args)
{

String str = "abcdcabcdacbdadbca";

    String findStr = "b";
    int lastIndex = 0;
    int count = 0;

    while (lastIndex != -1) {

     lastIndex = str.indexOf(findStr, lastIndex);

     if (lastIndex != -1) {
      count++;
      lastIndex += findStr.length();

     }
    }
    System.out.println(count);
}

}
```

Output:

```
4
```

Can we overload static methods in java

Posted by: instance of java Posted date: **19:20** / comment : 0

- Yes. We can overload static methods in java.
- Method overriding is not possible but method overloading is possible for static methods.
- Before that lets see about method overloading in java.

Method overloading:

- Defining multiple methods with same name and with different arguments is known as method overloading.
- Multiple methods with same name and different arguments so compile time itself we can tell which method is going to get executed based on method call.
- Method overloading also known as compile time polymorphism.

Static methods - method overloading

- Its always possibles static method overloading.
- Defining multiple static methods with same name and different arguments will possible.
- By this we can define multiple main methods in our class with different arguments but only default main method will be called by the JVM remaining methods we need to call explicitly.
- Lets see an example java program which explains static method overloading.

```
class StaticMethodOverloading{

public static void staticMethod(){

System.out.println("staticMethod(): Zero arguments");

}

public static void staticMethod(int a){

System.out.println("staticMethod(int a): one argument");

}

public static void staticMethod(String str, int x){

```

```
System.out.println("staticMethod(String str, int x): two arguments");

}

public static void main(String []args){

  StaticMethodOverloading.staticMethod();
  StaticMethodOverloading.staticMethod(12);
  StaticMethodOverloading.staticMethod("Static method overloading",10);

}
}
```

Output:

```
staticMethod(): Zero arguments
staticMethod(int a): one argument
staticMethod(String str, int x): two arguments
```

Java Program to overload main method in java

```
class mainMethodOverloading{

public static void main(boolean x){

System.out.println("main(boolean x) called ");

}

public static void main(int x){

System.out.println("main(int x) called");

}

public static void main(int a, int b){

System.out.println("main(int a, int b) called");

}

public static void main(String []args){


System.out.println("main(String []args) called ");

```

```
 mainMethodOverloading.main(true);
 mainMethodOverloading.main(10);
mainMethodOverloading.main(37,46);


}
}
```

Output:

```
main(String []args) called
main(boolean x) called
main(int x) called
main(int a, int b) called
```

Can we override static methods in java

Posted by: instance of java Posted date: **16:58** / comment : 0

- Exact answer is NO. We can not override static methods.
- Before discussing this topic lets see what is static in java.

Static methods in java:

- Static means class level if we declare any data or method as static then those data(variables) and methods belongs to that class at class level.
- Static variables and static methods belongs to class level.
- Static methods are not the part of objects state . Belongs to class
- Without using object we can call these static methods.

here the detailed explanation on static methods

Instance methods in java:

- Instance methods will be called at run time.
- Instance variables and instance methods are object level. variables values may change from one object to another where as static variable values are class level so if we access by using any object and changes that values will effect to all objects means its class level variable.
- Lets see about overriding in java.

Method overriding in java:

- Defining the super class method in sub class with same signature.

- Even though in inheritance all properties of supers class can access in sub class we have an option of overriding super class methods in sub class known as method overriding.
- So by this every-time sub-most object method will be called.

Instance methods - Method Overriding

```
class SuperClassDemo{

public void instanceMethod(){

System.out.println("SuperClassDemo instanceMethodcalled");

}

}
```

```
class SubClassDemo extends SuperClassDemo{

public void instanceMethod(){

System.out.println("SubClassDemo instanceMethod called");

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

 SuperClassDemo superObj= new SuperClassDemo();
 SuperClassDemo  superobj1= new  SubClassDem();
 SubClassDemo subObj= new  SubClassDem();
 // here no need to create object to call a static method please note that.

 superObj.instanceMethod();
 superObj1.instanceMethod();
 subObj.instanceMethod();

}
}
```

Output

```
SuperClassDemo instanceMethodcalled
SubClassDemo instanceMethodcalled
SubClassDemo instanceMethodcalled
```

Static methods - method overriding:

- Lest an example program on static methods in method overriding concept and then discus about this clearly.

```
class SuperClassDemo{

public static void staticMethod(){

System.out.println("SuperClassDemo staticMethod called");

}

}
```

```
class SubClassDemo extends SuperClassDemo{

public static void staticMethod(){

System.out.println("SubClassDemo staticMethod called");

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

 SuperClassDemo superObj= new SuperClassDemo();
 SuperClassDemo  superobj1= new  SubClassDem();
 SubClassDemo subObj= new  SubClassDem();
 // here no need to create object to call a static method please note that.

 superObj.staticMethod();
 superObj1.staticMethod();
 subObj.staticMethod();

}
}
```

Output

```
SuperClassDemo staticMethod called
SuperClassDemo staticMethod called
SubClassDemo staticMethod called
```

Can we call sub class methods using super class object

Posted by: instance of java Posted date: **16:23** / comment : 0

- Common coding interview question everybody facing in interviews is can we call sub class method using super class object? or can we call child class method using parent class? or can a super class call a subclass' method.
- The answer is No. but still we can say yes. so we need to what are all those scenarios of calling sub class method from super class and lets discuss about which is the actual possible case to call sub class method using super class object.
- Before that let me explain what is inheritance and how the super and sub classes related each other.

Inheritance:

- Getting the properties from one class object to another class object is known as inheritance.
- Two types of inheritance
- Getting the properties from one class object to another with level wise and with some priorities is known as multilevel inheritance.
- Getting the properties from one class object to another class object with same priority is known as multiple inheritance
- Java does not supports multiple inheritance

Read this:

Why Java does not supports multiple inheritance

Creating super class object and calling methods

```
//Multilevel inheritance program
Class A{

int a;

public void print(){

System.out.println("print method of super class A");

}

}
Class B extends A{

public void show(){

 System.out.println("show method of sub class B");

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

A obj= new A();
obj.a=10;

obj.print();
 //obj.show(); // compile time error

 System.out.println("a="obj.a);

}
}
```

Output:

```
print method of super class  A
10
```

- In above program super class and sub class is there and sub class extending super class.
- But we created object for super class so we can call only super class methods on that object.
- If we create sub class object we can call super class and sub class methods on that object now we can able to access super class A methods only.
- So by creating super class object we can call only super class methods

Creating super class object and calling methods

```
//Multilevel inheritance program
Class A{

int a;

public void print(){

System.out.println("print method of super class A");

}

}
Class B extends A{

public void show(){

 System.out.println("show method of sub class B");

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

B obj= new B();
obj.a=10;

obj.print();
obj.show();

 System.out.println("a="obj.a);

}
}
```

Output:

```
print method of super class  A
show method of sub class B
10
```

- Above program shows sub class object using super class methods and variables as we said in inheritance concept all super class members can accessed by the sub class means all super class members are available to sub class if sub class extending super class.
- So whenever we create object of sub class it will call sub class constructor and from sub class constructor super class constructor will be called so memory will be allocated to all super class non static member and sub class non static members.
- So we can call super class methods using sub class.
- B obj= new B();
- obj.print();
- So by creating sub class object we can call both super class methods and sub class methods.

Assigning sub class reference to super class object.

```
//Multilevel inheritance program
Class A{

int a;

public void print(){

System.out.println("print method of super class A");

}

}
Class B extends A{
```

```

public void show(){

 System.out.println("show method of sub class B");

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

A obj= new B();
obj.a=10;

obj.print();
//obj.show(); // compile time error

 System.out.println("a="obj.a);

}
}
```

Output:

```
print method of super class  A
10
```

- Yes we can assign sub class object to super class reference.
- So here Sub class object is created so memory allocated to all super class members
- so can we call all methods ? No eventhough sub class object is created we can not call sub class methods because we assigned sub class object to super class reference.
- Then how its possible to call sub class methods?
- Yes its possible to call sub class methods using super class by type casting to sub class object .
- By type casting super class object to sub class object we can access all corresponding sub class and all super class methods on that reference.

Assigning sub class reference to super class object.

```
//Multilevel inheritance program
Class A{

int a;

public void print(){

System.out.println("print method of super class A");

}

}
Class B extends A{

public void show(){

 System.out.println("show method of sub class B");

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

A obj= new B();
obj.a=10;

obj.print();
//obj.show();   compile time error
((B)obj).show(); // works fine


 System.out.println("a="obj.a);

}
}
```

Output:

```
print method of super class  A
show method of sub class B
10
```

Can we override private method in java

Posted by: instance of java Posted date: **17:20** / comment : 0

- We can not override private methods in java.
- The basic inheritance principle is when we are extending a class we can access all non private members of a class so private members are private to that class only we can not access anywhere outside the class if we declare anything as private.
- Know more information about access specifiers here

```
Class A{

  int a;
 private int b;

public void print(){

System.out.println("print method of super class A");

}

private void add(){

System.out.println("add method of super class A");

}
}
Class B extends A{

public void show(){

 System.out.println("show method of sub class B");

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

B obj= new B();
obj.a=30;

obj.print();
obj.show();

 System.out.println("a="obj.a);
//obj.b=20;  compile time error. The field Super.a is not visible
//obj.add(); compile time error : The method show() from the type Super is not visible

}
}
```

Output:

```
print method of super class  A
show method of sub class B
30
```

- From the above program we can say super class private members are not accessible to sub class.
- lets see what will happen if we try to override a method of super class?
- Sorry if we are unable to access super class private methods then there should not be a questions of overriding that method right?
- Yes private methods of super class can not be overridden in sub class.
- Even if we try to override same method of super class that will became a sub class own method not overridden method.

```
Class A{

  int a;
 private int b;

private void add(){

System.out.println("add method of super class A");

}
}
Class B extends A{

private void add(){

 System.out.println("add method of sub class B");

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

B obj= new B();
obj.a=30;

obj.add();


 System.out.println("a="obj.a);

}
}
```

Output:

```
add method of sub class B
30
```

- We can prove this by providing @override annotation in sub class method

```
Class A{

  int a;
 private int b;

private void add(){

```

```
System.out.println("add method of super class A");

}
}
Class B extends A{
 //@override
private void add(){// if we place @override annotation compile time error will come here

 System.out.println("add method of sub class B");

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

B obj= new B();
obj.a=30;

obj.add();


 System.out.println("a="obj.a);

}
}
```

Output:

```
add method of sub class B
30
```