1.program to create and throw exceptionlass InvalidAgeException extends Exception

{

public InvalidAgeException (String str)

{

// calling the constructor of parent Exception

super(str);

}

}

// class that uses custom exception InvalidAgeException

public class TestCustomException1

{

// method to check the age

static void validate (int age) throws InvalidAgeException{

if(age < 18){

// throw an object of user defined exception

throw new InvalidAgeException("age is not valid to vote");

}

else {

System.out.println("welcome to vote");

}

}

// main method

public static void main(String args[])

{

try

{

// calling the method

validate(13);

}

catch (InvalidAgeException ex)

{

System.out.println("Caught the exception");

// printing the message from InvalidAgeException object

System.out.println("Exception occured: " + ex);

}

System.out.println("rest of the code...");

}

}

Class InvalidAgeException extends Exception

{

public InvalidAgeException (String str)

{

super(str);

}

}

public class TestCustomException1

{

static void validate (int age) throws InvalidAgeException{

if(age < 18){

throw new InvalidAgeException("age is not valid to vote");

}

else {

System.out.println("welcome to vote");

}

}

public static void main(String args[])

{

try

{

validate(13);

}

catch (InvalidAgeException ex)

{

System.out.println("Caught the exception");

System.out.println("Exception occured: " + ex);

}

System.out.println("rest of the code...");

}

}

2. How do we a reverse a string?

Since the strings are immutable objects, you need to create another string to reverse them. The string class doesn't have a reverse method to reverse the string. It has a toCharArray() method to do the reverse.

By Using toCharArray()

.By Using StringBuilder

StringBuilder or StringBuffer class has an in-build method reverse() to reverse the characters in the string. This method replaces the sequence of the characters in reverse order. The reverse method is the static method that has the logic to reverse a string in Java.

By Using While Loop or For Loop

Simply handle the string within the while loop or the for loop. Get the length of the string with the help of a cursor move or iterate through the index of the string and terminate the loop.

By Using StringBuffer

The String class requires a reverse() function, hence first convert the input string to a StringBuffer, using the StringBuffer method. Then use the reverse() method to reverse the string.

3.program that detects the duplicate character in a string ?

public class DuplicateCharacters {

public static void main(String[] args) {

String string1 = "Great responsibility";

int count;

char string[] = string1.toCharArray();

System.out.println("Duplicate characters in a given string: ");

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

count = 1;

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

if(string[i] == string[j] && string[i] != ' ') {

count++;

string[j] = '0';

import java.util.ArrayList;

import java.util.Arrays;

import java.util.LinkedHashSet;

import java.util.Set;

class Main {

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

ArrayList<Integer> numbers = new ArrayList<>(Arrays.asList(1, 2, 3, 4, 1, 3));

System.out.println("ArrayList with duplicate elements: " + numbers);

Set<Integer> set = new LinkedHashSet<>();

Set.addAll(Numbers);

numbers.clear();

numbers.addAll(set);

System.out.println("ArrayList without duplicate elements: " + numbers);

}

}

4.program to remove duplicates in a array list ?

import java.util.ArrayList;

import java.util.Arrays;

import java.util.LinkedHashSet;

import java.util.Set;

class Main {

public static void main(String[] args) {

ArrayList<Integer> numbers = new ArrayList<>(Arrays.asList(1, 2, 3, 4, 1, 3));

System.out.println("ArrayList with duplicate elements: " + numbers);

Set<Integer> set = new LinkedHashSet<>();

set.addAll(numbers);

numbers.clear();

numbers.addAll(set);

System.out.println("ArrayList without duplicate elements: " + numbers);

}

}

5.infinite loop :

An infinite loop in Java is a sequence of instructions that loops indefinitely unless the system crashes. Infinite loops in Java occur when the terminating condition of the loop is not met. Usually, an infinite loop in Java is a programming error, but sometimes infinite loop in Java is also used intentionally, for example in a wait condition.

6.Program to demonstrate method overloading by changing data types ?

class Adder{

static int add(int a, int b){return a+b;}

static double add(double a, double b){return a+b;}

}

class TestOverloading2{

public static void main(String[] args){

System.out.println(Adder.add(11,11));

System.out.println(Adder.add(12.3,12.6));

}}

7.Program to demonstrate method overloading by changing a number of arguments ?

class Adder{

static int add(int a,int b){return a+b;}

static int add(int a,int b,int c){return a+b+c;}

}

class TestOverloading1{

public static void main(String[] args){

System.out.println(Adder.add(11,11));

System.out.println(Adder.add(11,11,11));

}}

8`How to read a file in java ?

Here are multiple ways of writing and reading a text file. this is required while dealing with many applications. There are several ways to read a plain text file in Java e.g. you can use FileReader, BufferedReader, or Scanner to read a text file. Every utility provides something special e.g. BufferedReader provides buffering of data for fast reading, and Scanner provides parsing ability.

Methods:

Using BufferedReader class

Using Scanner class

Using File Reader class

Reading the whole file in a List

Read a text file as Striing

8. what is get name & example ?

getName() returns the name of the entity (class, interface, array class, primitive type, or void) represented by this Class object, as a String.

package com.tutorialspoint;

import java.lang.\*;

public class ClassDemo {

public static void main(String[] args) {

ClassDemo cl = new ClassDemo();

Class c1Class = cl.getClass();

String name = c1Class.getName();

System.out.println("Class Name = " + name);

}

}

11.How to convert string to data in java ?

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.time.Instant;

import java.time.LocalDate;

import java.time.Period;

import java.time.ZoneId;

import java.time.ZonedDateTime;

import java.util.Date;

import java.util.Scanner;

public class CalculatingAge {

public static Date StringToDate(String dob) throws ParseException{

SimpleDateFormat formatter = new SimpleDateFormat("dd-MM-yyyy");

Date date = formatter.parse(dob);

System.out.println("Date object value: "+date);

return date;

}

public static void main(String args[]) throws ParseException {

Scanner sc = new Scanner(System.in);

System.out.println("Enter your name: ");

String name = sc.next();

System.out.println("Enter your date of birth (dd-MM-yyyy): ");

String dob = sc.next();

Date date = CalculatingAge.StringToDate(dob);

Instant instant = date.toInstant();

ZonedDateTime zone = instant.atZone(ZoneId.systemDefault());

LocalDate givenDate = zone.toLocalDate();

Period period = Period.between(givenDate, LocalDate.now());

System.out.print("Hello "+name+" your current age is: ");

System.out.print(period.getYears()+" years "+period.getMonths()+" and "+period.getDays()+" days");

}

}

Output:

Enter your name:

Krishna

Enter your date of birth (dd-MM-yyyy):

26-09-1989

Date object value: Tue Sep 26 00:00:00 IST 1989

Hello Krishna your current age is: 29 years 8 and 5 days

12.prgm to generate the pattern:

public class Pattern

{

public static void main(String args[])

{

int i, j, row=6;

for(i=0; i<row; i++)

{

for(j=0; j<=i; j++)

{

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

}

System.out.println();

}

}

}

Output

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

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

13.How to find duplicate character in a string :

ublic class DuplicateCharacters {

public static void main(String[] args) {

String string1 = "Great responsibility";

int count;

char string[] = string1.toCharArray();

System.out.println("Duplicate characters in a given string: ");

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

count = 1;

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

if(string[i] == string[j] && string[i] != ' ') {

count++;

string[j] = '0';

}

}

if(count > 1 && string[i] != '0')

System.out.println(string[i]);

}

}

}

Output:

Duplicate characters in a given string:

r

e

t

s

i

14.remove special character from a string:

public class RemoveSpecialCharacterExample2

{

public static void main(String args[])

{

String str = "Hello+-^Java+ -Programmer^ ^^-- ^^^ +!";

str = str.replaceAll("[-+^]\*", " ");

//str=str.replaceAll("\\W", " ")

System.out.println(str);

}

}

Output

Hello Java Programmer

16.how to sort array :

The sorting is a way to arrange elements of a list or array in a certain order. The order may be in ascending or descending order. The numerical and lexicographical (alphabetical) order is a widely used order. The ascending order arranges the elements in the lowest to highest order. It is also known as natural order or numerical order. We can perform sorting in the following ways:

Using the sort() Method

Without using the method

Using the for Loop

Using the User Defined Method

Eg:

import java.util.Arrays;

public class Sorting {

public static void main (String [] args) {

int [] array = {45,12,85,32,89,39,69,44,42,1,6,8};

Arrays.sort(array);

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

System.out.println(array[i]);

};

}

}

17.program to do bubble sort:

public class BubbleSortExample {

static void bubbleSort(int[] arr) {

int n = arr.length;

int temp = 0;

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

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

if(arr[j-1] > arr[j]){

//swap elements

temp = arr[j-1];

arr[j-1] = arr[j];

arr[j] = temp;

}

}

}

}

public static void main(String[] args) {

int arr[] ={3,60,35,2,45,320,5};

System.out.println("Array Before Bubble Sort");

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

System.out.print(arr[i] + " ");

}

System.out.println();

bubbleSort(arr);

System.out.println("Array After Bubble Sort");

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

System.out.print(arr[i] + " ");

}

}

}

Output:

…Array Before Bubble Sort

3 60 35 2 45 320 5

Array After Bubble Sort

2 3 5 35 45 60 3

18`pgm that sorts hashmap in integer:

import java.util.Collections;

import java.util.Comparator;

import java.util.HashMap;

import java.util.LinkedHashMap;

import java.util.LinkedList;

import java.util.List;

import java.util.Map;

import java.util.Map.Entry;

public class SortByValue

{

Map<String, Integer> map = new HashMap<String, Integer>();

public static void main(String[] args)

{

SortByValue sv = new SortByValue();

sv.createMap();

System.out.println("Sorting values in ascending order:");

sv.sortByValue(true);

System.out.println("Sorting values in descending order");

sv.sortByValue(false);

}

void createMap()

{

map.put("Apple", 65000);

map.put("HP", 20000);

map.put("Dell", 32000);

map.put("Asus", 21478);

map.put("Samsung", 36546);

map.put("Lenovo", 19990);

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

printMap(map);

}

void sortByValue(boolean order)

{

List<Entry<String, Integer>> list = new LinkedList<Entry<String, Integer>>(map.entrySet())

Collections.sort(list, new Comparator<Entry<String, Integer>>()

{

public int compare(Entry<String, Integer> o1, Entry<String, Integer> o2)

{

if (order)

{

//compare two object and return an integer

return o1.getValue().compareTo(o2.getValue());}

else

{

return o2.getValue().compareTo(o1.getValue());

}

}

});

Map<String, Integer> sortedMap = new LinkedHashMap<String, Integer>();

for (Entry<String, Integer> entry : list)

{

sortedMap.put(entry.getKey(), entry.getValue());

}

printMap(sortedMap);

public void printMap(Map<String, Integer> map)

{

System.out.println("Company\t Price ");

for (Entry<String, Integer> entry : map.entrySet())

{

System.out.println(entry.getKey() +"\t"+entry.getValue());

}

System.out.println("\n");

}

}

Output:

Before sorting:

Company Price

Dell 32000

HP 20000

Lenovo 19990

Samsung 36546

Apple 65000

Asus 21478

Sorting values in ascending order:

Company Price

Lenovo 19990

HP 20000

Asus 21478

Dell 32000

Samsung 36546

MAC Book 65000

Sorting values in descending order:

Company Price

MAC Book 65000

Samsung 36546

Dell 32000

Asus 21478

HP 20000

Lenovo 19990

20`pgm to show null pointer exception

import java.util.UUID;

import java.io.\*;

class Singleton

{

private static Singleton single = null;

private String ID = null;

private Singleton()

{

ID = UUID.randomUUID().toString();

}

public static Singleton getInstance()

{

if (single == null)

single = new Singleton();

return single;

}

public String getID()

{

return this.ID;

}

public class TestSingleton

{

public static void main(String[] args)

{

Singleton s = Singleton.getInstance();

System.out.println(s.getID());

} }

…