Assignment 1: Manipulation of strings

**Problem Description:** Write a program in Java which accepts 2 strings as command line arguments. The program should do the following:

class A

{

 public static void main(String args[])

 {

  System.out.println(args[0] + “ technologies ”);

  System.out.println(args[1]);

   }

}

{

  public static void main(String args[])

  {

  int a=Integer.parseInt(args[1]);//"10" convert to 10 and it will store in a

  int b=Integer.parseInt(args[0]);//"20" convert to 20 and it will store in b

   System.out.println(a+1);

   System.out.println(b+1);

  }

}

Assignment 2: Working with dates

import static java.util.Calendar.DATE;

import static java.util.Calendar.DAY\_OF\_WEEK;

import static java.util.Calendar.MONTH;

import static java.util.Calendar.YEAR;

import java.text.DateFormatSymbols;

import java.util.GregorianCalendar;

public class MainClass {

public static void main(String[] args) {

GregorianCalendar birthdate = new GregorianCalendar(1999, 1, 1);

GregorianCalendar today = new GregorianCalendar(); // Today's date

GregorianCalendar birthday = new GregorianCalendar(today.get(YEAR), birthdate.get(MONTH),

birthdate.get(DATE));

int age = today.get(today.YEAR) - birthdate.get(YEAR);

String[] weekdays = new DateFormatSymbols().getWeekdays(); // Get day names

System.out.println("You were born on a " + weekdays[birthdate.get(DAY\_OF\_WEEK)]);

System.out.println("This year you " + (birthday.after(today) ? " will be " : "are ") + age

+ " years old.");

System.out.println("In " + today.get(YEAR) + " your birthday "

+ (today.before(birthday) ? "will be" : "was") + " on a "

+ weekdays[birthday.get(DAY\_OF\_WEEK)] + ".");

}

}

Assignment 3: Hash Table

Store the names of 10 major cities and the names of corresponding countries in a hash table. Accept the name of a city as a command-line argument and display the country in which it is situated. Make provision to display a message if the user either forgets to provide command-line argument or specifies a city that is not in the hash table.

Map<String, ArrayList<String>> countries\_ht = new HashMap();

Map<String, ArrayList<String>> states\_ht = new HashMap();

List<String> states = countries\_ht.get("USA")

List<String> cities = states\_ht.get("California")

Map<String, String> city\_to\_state\_rlt = new HashMap();

Map<String, String> state\_to\_county\_rlt = new HashMap();

String state = city\_to\_state\_rlt.get("Los Angeles"); // returns "California

String country = state\_to\_county\_rlt.get(state); // returns USA

Assignment 4: Working with collection classes

**:** Take in 10 numbers as command line arguments and store it in a collection. The numbers are to be displayed in the reverse order in which they were entered. Proper error messages should be displayed if:

1. command line arguments have not been entered
2. less than 10 numbers have been fed in
3. If one of the arguments is not a valid number

public class Add {

public static void main(String[] args) {

int sum = 0;

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

sum = sum + Integer.parseInt(args[i]);

}

System.out.println("The sum of the arguments passed is " + sum);

}

}

List<String> listStrings = new ArrayList<String>();

listStrings.add("One");

listStrings.add("Two");

listStrings.add("Three");

listStrings.add("Four");

System.out.println(listStrings);

List<String> listStrings = new LinkedList<String>();

listStrings.add("Five");

listStrings.add("Six");

listStrings.add("Seven");

listStrings.add("Eight");

System.out.println(listStrings);

List<Object> listAnything = new ArrayList<Object>();

List<String> listWords = new ArrayList<String>();

List<Integer> listNumbers = new ArrayList<Integer>();

List<String> linkedWords = new LinkedList<String>();

List<Number> linkedNumbers = new LinkedList<>();

linkedNumbers.add(new Integer(123));

linkedNumbers.add(new Float(3.1415));

linkedNumbers.add(new Double(299.988));

linkedNumbers.add(new Long(67000));

LinkedList<Number> numbers = new LinkedList<Number>();

// add elements to the list...

// get the first and the last elements:

Number first = numbers.getFirst();

Number last = numbers.getLast();

if (listStrings.remove("Ten")) {

    System.out.println("Removed");

} else {

    System.out.println("There is no such element");

}

if (listStrings.contains("Hello")) {

    System.out.println("Found the element");

} else {

    System.out.println("There is no such element");

}

int firstIndex = linkedNumbers.indexOf(1234);

int lastIndex = listStrings.indexOf("Hello");

List<String> sourceList = new ArrayList<String>();

sourceList.add("A");

sourceList.add("B");

sourceList.add("C");

sourceList.add("D");

List<String> destList = new ArrayList<String>();

destList.add("V");

destList.add("W");

destList.add("X");

destList.add("Y");

destList.add("Z");

System.out.println("destList before copy: " + destList);

Collections.copy(destList, sourceList);

System.out.println("destList after copy: " + destList);