Mica Ella G. Galindes

BSIT 1A

**COMPUTER PROGRAMMING 2**

ACTIVITY 1

1. **Counting Duplicate Characters.**

public class CP2 {

static final int NO\_OF\_CHARS = 256;

static void countDuplicateChar(String str,

int[] count)

{

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

count[str.charAt(i)]++;

}

static void printDuplicateChar(String str)

{

int count[] = new int[NO\_OF\_CHARS];

countDuplicateChar(str, count);

for (int i = 0; i < NO\_OF\_CHARS; i++)

if (count[i] > 1)

System.out.println((char)(i) +

", count = " + count[i]);

}

public static void main(String[] args)

{

String str = "computerprogramming";

printDuplicateChar(str);

}

}

1. **Finding the first non-repeated character.**

public class CP2 {

static final int NO\_OF\_CHARS = 256;

static char count[] = new char[NO\_OF\_CHARS];

static void getNonRepeatingArray(String str)

{

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

count[str.charAt(i)]++;

}

static int firstNonRepeatedChar(String str)

{

getNonRepeatingArray(str);

int index = -1, i;

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

if (count[str.charAt(i)] == 1) {

index = i;

break;

}

}

return index;

}

public static void main(String[] args)

{

String str = "computerprogramming";

int index = firstNonRepeatedChar(str);

System.out.println(

index == -1

? "Either all characters are repeating or string " + "is empty"

: "First non-repeating character is " + str.charAt(index));

}

}

1. **Checking whether a string contains only digits.**

class CP2 {

public static boolean

onlyDigits(String str, int n)

{

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

if (str.charAt(i) >= '0'&& str.charAt(i) <= '9') {

return true;

}

else {

return false;

}

}

return false;

}

public static void main(String args[])

{

String str = "1234";

int len = str.length();

System.out.println(onlyDigits(str, len));

}

}

1. **Removing white space from a string.**

class CP2 {

static int removeWhiteSpaces(char []str)

{

int count = 0;

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

if (str[i] != ' ')

str[count++] = str[i];

return count;

}

public static void main(String[] args)

{

char str[] = "c o m p u t e r p r o g r a m mi n g".toCharArray();

int i = removeWhiteSpaces(str);

System.out.println(String.valueOf(str).subSequence(0, i));

}

}

1. **Checking whether two strings are anagram.**

import java.io.\*;

import java.util.\*;

class CP2 {

static int NO\_OF\_CHARS = 256;

static boolean areAnagram(char str1[], char str2[])

{

int count1[] = new int[NO\_OF\_CHARS];

Arrays.fill(count1, 0);

int count2[] = new int[NO\_OF\_CHARS];

Arrays.fill(count2, 0);

int i;

for (i = 0; i < str1.length && i < str2.length;

i++) {

count1[str1[i]]++;

count2[str2[i]]++;

}

if (str1.length != str2.length)

return false;

for (i = 0; i < NO\_OF\_CHARS; i++)

if (count1[i] != count2[i])

return false;

return true;

}

public static void main(String args[])

{

char str1[] = ("computerprogramming").toCharArray();

char str2[] = ("programmingcomputer").toCharArray();

if (areAnagram(str1, str2))

System.out.println("The two strings are" + " anagram of each other");

else

System.out.println("The two strings are not" + " anagram of each other");

}

}