**Level 1 practice problem of String**

**-----------------------------------------------------------------------------------------------------------------------------**

**Program 1: Compare Two Strings Using charAt() and equals()**

import java.util.Scanner;

public class CompareStrings {

public static boolean compareUsingCharAt(String s1, String s2) {

if (s1.length() != s2.length()) return false;

for (int i = 0; i < s1.length(); i++) {

if (s1.charAt(i) != s2.charAt(i)) return false;

}

return true;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter first string: ");

String str1 = scanner.next();

System.out.print("Enter second string: ");

String str2 = scanner.next();

boolean resultCharAt = compareUsingCharAt(str1, str2);

boolean resultEquals = str1.equals(str2);

System.out.println("Result using charAt(): " + resultCharAt);

System.out.println("Result using equals(): " + resultEquals);

}

}

**Program 2: Create Substring Using charAt() and Compare with substring()**

import java.util.Scanner;

public class SubstringComparison {

public static String substringUsingCharAt(String str, int start, int end) {

String result = "";

for (int i = start; i < end && i < str.length(); i++) {

result += str.charAt(i);

}

return result;

}

public static boolean compareUsingCharAt(String s1, String s2) {

if (s1.length() != s2.length()) return false;

for (int i = 0; i < s1.length(); i++) {

if (s1.charAt(i) != s2.charAt(i)) return false;

}

return true;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter text: ");

String text = scanner.next();

System.out.print("Enter start index: ");

int start = scanner.nextInt();

System.out.print("Enter end index: ");

int end = scanner.nextInt();

String substrCustom = substringUsingCharAt(text, start, end);

String substrBuiltIn = text.substring(start, end);

System.out.println("Custom substring: " + substrCustom);

System.out.println("Built-in substring: " + substrBuiltIn);

System.out.println("Are they equal? " + compareUsingCharAt(substrCustom, substrBuiltIn));

}

}

**Program 3: Return All Characters Using Custom Method and Compare with toCharArray()**

import java.util.Scanner;

public class CharArrayComparison {

public static char[] getChars(String str) {

char[] result = new char[str.length()];

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

result[i] = str.charAt(i);

}

return result;

}

public static boolean compareCharArrays(char[] a, char[] b) {

if (a.length != b.length) return false;

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

if (a[i] != b[i]) return false;

}

return true;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a string: ");

String text = scanner.next();

char[] customChars = getChars(text);

char[] builtInChars = text.toCharArray();

System.out.println("Are both character arrays equal? " + compareCharArrays(customChars, builtInChars));

}

}

**Program 4: Demonstrate NullPointerException**

public class NullPointerDemo {

public static void generateException() {

String text = null;

System.out.println(text.length()); // causes NullPointerException

}

public static void handleException() {

try {

String text = null;

System.out.println(text.length());

} catch (NullPointerException e) {

System.out.println("Caught NullPointerException.");

}

}

public static void main(String[] args) {

// generateException(); // Uncomment to see unhandled exception

handleException();

}

}

**Program 5: Demonstrate StringIndexOutOfBoundsException**

import java.util.Scanner;

public class StringIndexDemo {

public static void generateException(String text) {

System.out.println(text.charAt(text.length())); // out of bounds

}

public static void handleException(String text) {

try {

System.out.println(text.charAt(text.length()));

} catch (StringIndexOutOfBoundsException e) {

System.out.println("Caught StringIndexOutOfBoundsException.");

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a string: ");

String input = scanner.next();

// generateException(input); // Uncomment to crash

handleException(input);

}

}

**Program 6: Demonstrate IllegalArgumentException**

import java.util.Scanner;

public class IllegalArgumentDemo {

public static void generateException(String text) {

System.out.println(text.substring(5, 2)); // illegal start/end

}

public static void handleException(String text) {

try {

System.out.println(text.substring(5, 2));

} catch (IllegalArgumentException e) {

System.out.println("Caught IllegalArgumentException.");

} catch (RuntimeException e) {

System.out.println("Caught RuntimeException.");

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a string: ");

String input = scanner.next();

// generateException(input); // Uncomment to crash

handleException(input);

}

}

**Program 7: Demonstrate NumberFormatException**

import java.util.Scanner;

public class NumberFormatDemo {

public static void generateException(String text) {

int number = Integer.parseInt(text);

System.out.println("Parsed number: " + number);

}

public static void handleException(String text) {

try {

int number = Integer.parseInt(text);

System.out.println("Parsed number: " + number);

} catch (NumberFormatException e) {

System.out.println("Caught NumberFormatException.");

} catch (RuntimeException e) {

System.out.println("Caught RuntimeException.");

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a string to convert to number: ");

String input = scanner.next();

// generateException(input); // Uncomment to crash

handleException(input);

}

}

**Program 8: Demonstrate ArrayIndexOutOfBoundsException**

import java.util.Scanner;

public class ArrayIndexDemo {

public static void generateException(String[] names) {

System.out.println(names[names.length]); // out of bounds

}

public static void handleException(String[] names) {

try {

System.out.println(names[names.length]);

} catch (ArrayIndexOutOfBoundsException e) {

System.out.println("Caught ArrayIndexOutOfBoundsException.");

} catch (RuntimeException e) {

System.out.println("Caught RuntimeException.");

}

}

public static void main(String[] args) {

String[] names = {"Alice", "Bob", "Charlie"};

// generateException(names); // Uncomment to crash

handleException(names);

}

}

**Program 9: Convert Text to Uppercase and Compare**

import java.util.Scanner;

public class UppercaseConverter {

public static String toUpper(String text) {

String result = "";

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

char c = text.charAt(i);

if (c >= 'a' && c <= 'z') {

result += (char) (c - 32);

} else {

result += c;

}

}

return result;

}

public static boolean compareUsingCharAt(String a, String b) {

if (a.length() != b.length()) return false;

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

if (a.charAt(i) != b.charAt(i)) return false;

}

return true;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter text: ");

String text = scanner.nextLine();

String customUpper = toUpper(text);

String builtInUpper = text.toUpperCase();

System.out.println("Custom uppercase: " + customUpper);

System.out.println("Built-in uppercase: " + builtInUpper);

System.out.println("Are they equal? " + compareUsingCharAt(customUpper, builtInUpper));

}

}

// Program 10: Convert Text to Lowercase and Compare

import java.util.Scanner;

public class LowercaseConverter {

public static String toLower(String text) {

String result = "";

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

char c = text.charAt(i);

if (c >= 'A' && c <= 'Z') {

result += (char) (c + 32);

} else {

result += c;

}

}

return result;

}

public static boolean compareUsingCharAt(String a, String b) {

if (a.length() != b.length()) return false;

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

if (a.charAt(i) != b.charAt(i)) return false;

}

return true;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter text: ");

String text = scanner.nextLine();

String customLower = toLower(text);

String builtInLower = text.toLowerCase();

System.out.println("Custom lowercase: " + customLower);

System.out.println("Built-in lowercase: " + builtInLower);

System.out.println("Are they equal? " + compareUsingCharAt(customLower, builtInLower));

}

}

Level 1 Practice Programs

1. Write a program to compare two strings using the ***charAt()*** method and check the result with the built-in String ***equals()*** method

**Hint =>**

1. Take user input using the  ***Scanner*** ***next()*** method for 2 String variables
2. Write a method to compare two strings using the ***charAt()*** method and return a boolean result
3. Use the String Built-In method to check if the results are the same and display the result
4. Write a program to create a substring from a String using the ***charAt()*** method. Also, use the String built-in method ***substring()*** to find the substring of the text. Finally Compare the the two strings and display the results

**Hint =>**

1. Take user input using the  ***Scanner*** ***next()*** method to take the String variable and also the start and the end index to get the substring from the given text
2. Write a method to create a substring from a string using the ***charAt()*** method with the string, start, and end index as the parameters
3. Write a method to compare two strings using the charAt() method and return a boolean result
4. Use the String built-in method substring() to get the substring and compare the two strings. And finally display the result
5. Write a program to return all the characters in a string using the user-defined method,  compare the result with the String built-in toCharArray() method, and display the result

**Hint =>**

1. Take user input using the  ***Scanner*** ***next()*** method to take the text into a String variable
2. Write a method to return the characters in a string without using the ***toCharArray()***
3. Write a method to compare two string arrays and return a boolean result
4. In the main() call the user-defined method and the String built-in ​​toCharArray() method, compare the 2 arrays, and finally display the result
5. Write a program to demonstrate NullPointerException.

**Hint =>**

1. Write a Method to generate the Exception. Here define the variable text and initialize it to null. Then call one of the String Method to generate the exception
2. Write the Method to demonstrate ***NullPointerException***. Here define the variable text and initialize it to null. Then write try catch block for handling the Exception while accessing one of the ***String*** method
3. From the main Firstly call the method to generate the Exception then refactor the code to call the method to handle the RuntimeException
4. Write a program to demonstrate ***StringIndexOutOfBoundsException***

**Hint =>**

1. Define a variable of type String and take user input to assign a value
2. Write a Method to generate the Exception. Access the index using charAt() beyond the length of the String. This will generate a runtime exception and abruptly stop the program.
3. Write the Method to demonstrate ***StringIndexOutOfBoundsException***. Access the index using ***charAt()*** beyond the length of the String. Then write try catch block for Exception while accessing the String method
4. From the main Firstly call the method to generate the Exception then call the method to handle the RuntimeException
5. Write a program to demonstrate ***IllegalArgumentException***

**Hint =>**

1. Define a variable of type String and take user input to assign a value
2. Write a Method to generate the Exception. Here use the ***subString()*** and set the start index to be greater than the end index. This will generate a runtime exception and abruptly stop the program.
3. Write the Method to demonstrate ***IllegalArgumentException***. Here use the ***subString()*** and set the start index to be greater than the end index. This will generate a runtime exception. Use the try-catch block to handle the ***IllegalArgumentException*** and the generic runtime exception
4. From the main Firstly call the method to generate the Exception then call the method to handle the RuntimeException
5. Write a program to demonstrate ***NumberFormatException***

**Hint =>**

1. Define a variable to take user input as a String
2. Use Integer.parseInt() to generate this exception. ***Integer.parseInt()*** is a built-in function in java.lang.Integer class to extract the number from text. In case the text does not contain numbers the method will throw NumberFormatException which is a runtime exception
3. Write a Method to generate the Exception. Use ***Integer.parseInt(text)*** to extract number from the text. This will generate a runtime exception and abruptly stop the program.
4. Write the Method to demonstrate ***NumberFormatException***. Use ***Integer.parseInt(text)*** to extract number from the text. This will generate a runtime exception. Use the try-catch block to handle the ***NumberFormatException*** as well as the generic runtime exception
5. From the main Firstly call the method to generate the Exception then call the method to handle the RuntimeException
6. Write a program to demonstrate ***ArrayIndexOutOfBoundsException***

**Hint =>**

1. Define a variable of array of names and take input from the user
2. Write a Method to generate the Exception. Here access index larger then the length of the array. This will generate a runtime exception and abruptly stop the program.
3. Write the Method to demonstrate ***ArrayIndexOutOfBoundsException***. Here access index larger then the length of the array. This will generate a runtime exception. Use the try-catch block to handle the ***ArrayIndexOutOfBoundsException*** and the generic runtime exception
4. From the main Firstly call the method to generate the Exception then call the method to handle the RuntimeException
5. Write a program to convert the complete text to uppercase and compare the results

**Hint =>**

1. Take user input using the  ***Scanner*** ***nextLine()*** method to take the complete text into a String variable
2. Write a method using the String built-in ***charAt()*** method to convert each character if it is lowercase to the uppercase. Use the logic ASCII value of 'a' is 97 and 'A' is 65 so the difference is 32, similarly ASCII value of 'b' is 98 and 'B' is 66 so the difference is 32, and so on
3. Write a method to compare two strings using the charAt() method and return a boolean result
4. In the main() use the String built-in method ***toUpperCase()*** to get the uppercase text and compare the two strings using the user-defined method. And finally display the result
5. Write a program to convert the complete text to lowercase and compare the results

**Hint =>**

1. Take user input using the  ***Scanner*** ***nextLine()*** method to take the complete text into a String variable
2. Write a method using the String built-in ***charAt()*** method to convert each character if it is uppercase to the lowercase. Use the logic ASCII value of 'a' is 97 and 'A' is 65 so the difference is 32, similarly ASCII value of 'b' is 98 and 'B' is 66 so the difference is 32, and so on
3. Write a method to compare two strings using the charAt() method and return a boolean result
4. In the main() use the String built-in method ***toLowerCase()*** to get the lowercase text and compare the two strings using the user-defined method. And finally display the result

Level2 practice Problem of String

**Program 1: Find Length Without length() Method**

import java.util.Scanner;

public class StringLengthFinder {

public static int findLength(String str) {

int count = 0;

try {

while (true) {

str.charAt(count);

count++;

}

} catch (Exception e) {

return count;

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter a string: ");

String str = sc.next();

int customLength = findLength(str);

int builtInLength = str.length();

System.out.println("Length (custom method): " + customLength);

System.out.println("Length (built-in method): " + builtInLength);

}

}

**✅ Program 2: Split Text into Words (Compare with split())**

import java.util.\*;

public class CustomSplitter {

public static int findLength(String str) {

int count = 0;

try {

while (true) {

str.charAt(count++);

}

} catch (Exception e) {

return count;

}

}

public static String[] customSplit(String text) {

int count = 0;

int[] spaceIndexes = new int[findLength(text)];

for (int i = 0; i < findLength(text); i++) {

if (text.charAt(i) == ' ') {

spaceIndexes[count++] = i;

}

}

String[] words = new String[count + 1];

int start = 0;

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

int end = (i == count) ? findLength(text) : spaceIndexes[i];

String word = "";

for (int j = start; j < end; j++) word += text.charAt(j);

words[i] = word;

start = end + 1;

}

return words;

}

public static boolean compareArrays(String[] a, String[] b) {

if (a.length != b.length) return false;

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

if (!a[i].equals(b[i])) return false;

}

return true;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter text: ");

String input = sc.nextLine();

String[] custom = customSplit(input);

String[] builtin = input.split(" ");

System.out.println("Custom Split: " + Arrays.toString(custom));

System.out.println("Built-in Split: " + Arrays.toString(builtin));

System.out.println("Are both equal? " + compareArrays(custom, builtin));

}

}

**✅ Program 3: Split Text and Return Words with Lengths (2D Array)**

import java.util.\*;

public class WordLengthArray {

public static int findLength(String str) {

int count = 0;

try {

while (true) str.charAt(count++);

} catch (Exception e) {

return count;

}

}

public static String[] customSplit(String text) {

int count = 0;

int[] spaceIndexes = new int[findLength(text)];

for (int i = 0; i < findLength(text); i++) {

if (text.charAt(i) == ' ') spaceIndexes[count++] = i;

}

String[] words = new String[count + 1];

int start = 0;

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

int end = (i == count) ? findLength(text) : spaceIndexes[i];

String word = "";

for (int j = start; j < end; j++) word += text.charAt(j);

words[i] = word;

start = end + 1;

}

return words;

}

public static String[][] wordsWithLength(String[] words) {

String[][] result = new String[words.length][2];

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

result[i][0] = words[i];

result[i][1] = String.valueOf(findLength(words[i]));

}

return result;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter a sentence: ");

String input = sc.nextLine();

String[] words = customSplit(input);

String[][] result = wordsWithLength(words);

System.out.println("Word\tLength");

for (String[] row : result) {

System.out.println(row[0] + "\t" + Integer.parseInt(row[1]));

}

}

}

**✅ Program 4: Find Shortest and Longest Word**

import java.util.\*;

public class MinMaxWordLength {

public static int findLength(String str) {

int count = 0;

try {

while (true) str.charAt(count++);

} catch (Exception e) {

return count;

}

}

public static String[] customSplit(String text) {

int count = 0;

int[] spaceIndexes = new int[findLength(text)];

for (int i = 0; i < findLength(text); i++) {

if (text.charAt(i) == ' ') spaceIndexes[count++] = i;

}

String[] words = new String[count + 1];

int start = 0;

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

int end = (i == count) ? findLength(text) : spaceIndexes[i];

String word = "";

for (int j = start; j < end; j++) word += text.charAt(j);

words[i] = word;

start = end + 1;

}

return words;

}

public static String[][] wordsWithLength(String[] words) {

String[][] result = new String[words.length][2];

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

result[i][0] = words[i];

result[i][1] = String.valueOf(findLength(words[i]));

}

return result;

}

public static String[] findMinMax(String[][] wordLen) {

int minLen = Integer.parseInt(wordLen[0][1]);

int maxLen = minLen;

String minWord = wordLen[0][0];

String maxWord = wordLen[0][0];

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

int len = Integer.parseInt(wordLen[i][1]);

if (len < minLen) {

minLen = len;

minWord = wordLen[i][0];

}

if (len > maxLen) {

maxLen = len;

maxWord = wordLen[i][0];

}

}

return new String[]{minWord, maxWord};

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter a sentence: ");

String input = sc.nextLine();

String[] words = customSplit(input);

String[][] wordLen = wordsWithLength(words);

String[] result = findMinMax(wordLen);

System.out.println("Shortest word: " + result[0]);

System.out.println("Longest word: " + result[1]);

}

}

**Program 5: Count Vowels and Consonants**

import java.util.Scanner;

public class VowelConsonantCounter {

public static String checkCharType(char c) {

if (c >= 'A' && c <= 'Z') c = (char)(c + 32); // Convert to lowercase

if (c >= 'a' && c <= 'z') {

if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {

return "Vowel";

} else {

return "Consonant";

}

}

return "Not a Letter";

}

public static int[] countVowelsConsonants(String str) {

int vowels = 0, consonants = 0;

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

try {

char c = str.charAt(i);

String type = checkCharType(c);

if (type.equals("Vowel")) vowels++;

else if (type.equals("Consonant")) consonants++;

} catch (Exception e) {

break;

}

}

return new int[]{vowels, consonants};

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter a string: ");

String str = sc.nextLine();

int[] count = countVowelsConsonants(str);

System.out.println("Vowels: " + count[0]);

System.out.println("Consonants: " + count[1]);

}

}

**✅ Program 6: Display Character Type - Vowel, Consonant, or Not a Letter**

import java.util.Scanner;

public class VowelConsonantTable {

public static String checkCharType(char c) {

if (c >= 'A' && c <= 'Z') c = (char)(c + 32);

if (c >= 'a' && c <= 'z') {

if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {

return "Vowel";

} else {

return "Consonant";

}

}

return "Not a Letter";

}

public static String[][] classifyCharacters(String str) {

int len = 0;

try {

while (true) str.charAt(len++);

} catch (Exception e) {}

String[][] result = new String[len][2];

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

char c = str.charAt(i);

result[i][0] = String.valueOf(c);

result[i][1] = checkCharType(c);

}

return result;

}

public static void displayTable(String[][] data) {

System.out.println("Char\tType");

for (String[] row : data) {

System.out.println(row[0] + "\t" + row[1]);

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter text: ");

String input = sc.nextLine();

String[][] result = classifyCharacters(input);

displayTable(result);

}

}

**✅ Program 7: Trim Leading & Trailing Spaces Using charAt()**

import java.util.Scanner;

public class StringTrimmer {

public static int[] trimIndices(String str) {

int start = 0;

int end = -1;

int len = 0;

try {

while (true) str.charAt(len++);

} catch (Exception e) {}

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

if (str.charAt(i) != ' ') {

start = i;

break;

}

}

for (int i = len - 1; i >= 0; i--) {

if (str.charAt(i) != ' ') {

end = i;

break;

}

}

return new int[]{start, end};

}

public static String customSubstring(String str, int start, int end) {

String result = "";

for (int i = start; i <= end; i++) {

result += str.charAt(i);

}

return result;

}

public static boolean compareStrings(String s1, String s2) {

int len1 = s1.length(), len2 = s2.length();

if (len1 != len2) return false;

for (int i = 0; i < len1; i++) {

if (s1.charAt(i) != s2.charAt(i)) return false;

}

return true;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter string with spaces: ");

String input = sc.nextLine();

int[] trimPos = trimIndices(input);

String trimmedByCustom = customSubstring(input, trimPos[0], trimPos[1]);

String trimmedByBuiltin = input.trim();

System.out.println("Trimmed (custom): [" + trimmedByCustom + "]");

System.out.println("Trimmed (built-in): [" + trimmedByBuiltin + "]");

System.out.println("Are they equal? " + compareStrings(trimmedByCustom, trimmedByBuiltin));

}

}

**✅ Program 8: Voting Eligibility Based on Age**

import java.util.Random;

import java.util.Scanner;

public class VotingEligibility {

public static int[] generateAges(int n) {

Random rand = new Random();

int[] ages = new int[n];

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

ages[i] = rand.nextInt(90) + 10; // Random 2-digit age (10-99)

}

return ages;

}

public static String[][] checkEligibility(int[] ages) {

String[][] result = new String[ages.length][2];

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

result[i][0] = String.valueOf(ages[i]);

if (ages[i] < 0) result[i][1] = "false";

else result[i][1] = (ages[i] >= 18) ? "true" : "false";

}

return result;

}

public static void displayTable(String[][] data) {

System.out.println("Age\tCan Vote");

for (String[] row : data) {

System.out.println(row[0] + "\t" + row[1]);

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter number of students: ");

int n = sc.nextInt();

int[] ages = generateAges(n);

String[][] table = checkEligibility(ages);

displayTable(table);

}

}