Level 1 Practice Programs

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

Hint =>

Take user input using the Scanner next() method for 2 String variables

Write a method to compare two strings using the charAt() method and return a boolean result

Use the String Built-In method to check if the results are the same and display the result

import java.util.Scanner;

public class StringCompare {

public static boolean compareUsingCharAt(String str1, String str2) {

if (str1.length() != str2.length()) {

return false;

}

for (int i = 0; i < str1.length(); i++) {

if (str1.charAt(i) != str2.charAt(i)) {

return false;

}

}

return true;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String string1 = sc.next();

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

String string2 = sc.next();

boolean resultCharAt = compareUsingCharAt(string1, string2);

boolean resultEquals = string1.equals(string2);

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

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

if (resultCharAt == resultEquals) {

System.out.println("Both methods return the same result.");

} else {

System.out.println("Methods return different results.");

}

sc.close();

}

}

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 =>

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

Write a method to create a substring from a string using the charAt() method with the string, start, and end index as the parameters

Write a method to compare two strings using the charAt() method and return a boolean result

Use the String built-in method substring() to get the substring and compare the two strings. And finally display the result

import java.util.Scanner;

public class SubstringCompare {

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

String result = "";

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

result += str.charAt(i);

}

return result;

}

public static boolean compareUsingCharAt(String str1, String str2) {

if (str1.length() != str2.length()) {

return false;

}

for (int i = 0; i < str1.length(); i++) {

if (str1.charAt(i) != str2.charAt(i)) {

return false;

}

}

return true;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the original string: ");

String original = sc.next();

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

int start = sc.nextInt();

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

int end = sc.nextInt();

String manualSubstring = createSubstringUsingCharAt(original, start, end);

String builtInSubstring = original.substring(start, end);

System.out.println("\nSubstring using charAt(): " + manualSubstring);

System.out.println("Substring using substring(): " + builtInSubstring);

boolean areEqual = compareUsingCharAt(manualSubstring, builtInSubstring);

if (areEqual) {

System.out.println("Both substrings are equal.");

} else {

System.out.println("Substrings are not equal.");

}

sc.close();

}

}

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 =>

Take user input using the Scanner next() method to take the text into a String variable

Write a method to return the characters in a string without using the toCharArray()

Write a method to compare two string arrays and return a boolean result

In the main() call the user-defined method and the String built-in ​​toCharArray() method, compare the 2 arrays, and finally display the result

import java.util.Scanner;

public class CharArrayCompare {

public static char[] getCharsUsingCharAt(String str) {

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

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

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

}

return chars;

}

public static boolean compareCharArrays(char[] arr1, char[] arr2) {

if (arr1.length != arr2.length) {

return false;

}

for (int i = 0; i < arr1.length; i++) {

if (arr1[i] != arr2[i]) {

return false;

}

}

return true;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String input = sc.next();

char[] customArray = getCharsUsingCharAt(input);

char[] builtInArray = input.toCharArray();

System.out.print("\nCharacters using user-defined method: ");

for (char c : customArray) {

System.out.print(c + " ");

}

System.out.print("\nCharacters using toCharArray(): ");

for (char c : builtInArray) {

System.out.print(c + " ");

}

boolean areEqual = compareCharArrays(customArray, builtInArray);

System.out.println("\n\nArrays are equal: " + areEqual);

sc.close();

}

}

Write a program to demonstrate NullPointerException.

Hint =>

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

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

From the main Firstly call the method to generate the Exception then refactor the code to call the method to handle the RuntimeException

public class NullPointerDemo {

public static void generateException() {

String text = null;

System.out.println("Length of text: " + text.length()); // This will throw NullPointerException

}

public static void handleException() {

String text = null;

try {

System.out.println("Length of text: " + text.length());

} catch (NullPointerException e) {

System.out.println("Caught NullPointerException: " + e.getMessage());

}

}

public static void main(String[] args) {

System.out.println("Generating Exception:");

try {

generateException();

} catch (Exception e) {

System.out.println("Exception caught in main: " + e);

}

System.out.println("\nHandling Exception:");

handleException();

}

}

Write a program to demonstrate StringIndexOutOfBoundsException

Hint =>

Define a variable of type String and take user input to assign a value

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.

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

From the main Firstly call the method to generate the Exception then call the method to handle the RuntimeException

import java.util.Scanner;

public class StringIndexDemo {

public static void generateException(String text) {

System.out.println("Character at invalid index: " + text.charAt(text.length())); // Out of bounds

}

public static void handleException(String text) {

try {

System.out.println("Character at invalid index: " + text.charAt(text.length()));

} catch (StringIndexOutOfBoundsException e) {

System.out.println("Caught StringIndexOutOfBoundsException: " + e.getMessage());

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String input = sc.next();

System.out.println("\nGenerating Exception:");

try {

generateException(input);

} catch (Exception e) {

System.out.println("Exception caught in main: " + e);

}

System.out.println("\nHandling Exception:");

handleException(input);

sc.close();

}

}

Write a program to demonstrate IllegalArgumentException

Hint =>

Define a variable of type String and take user input to assign a value

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.

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

From the main Firstly call the method to generate the Exception then call the method to handle the RuntimeException

import java.util.Scanner;

public class IllegalArgumentDemo {

public static void generateException(String text) {

System.out.println("Substring (invalid): " + text.substring(5, 2)); // start > end

}

public static void handleException(String text) {

try {

System.out.println("Substring (invalid): " + text.substring(5, 2));

} catch (IllegalArgumentException e) {

System.out.println("Caught IllegalArgumentException: " + e.getMessage());

} catch (RuntimeException e) {

System.out.println("Caught RuntimeException: " + e.getMessage());

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String input = sc.next();

System.out.println("\nGenerating Exception:");

try {

generateException(input);

} catch (Exception e) {

System.out.println("Exception caught in main: " + e);

}

System.out.println("\nHandling Exception:");

handleException(input);

sc.close();

}

}

Write a program to demonstrate NumberFormatException

Hint =>

Define a variable to take user input as a String

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

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.

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

From the main Firstly call the method to generate the Exception then call the method to handle the RuntimeException

import java.util.Scanner;

public class NumberFormatDemo {

public static void generateException(String text) {

int number = Integer.parseInt(text); // This will throw NumberFormatException if text is non-numeric

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: " + e.getMessage());

} catch (RuntimeException e) {

System.out.println("Caught RuntimeException: " + e.getMessage());

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String input = sc.next();

System.out.println("\nGenerating Exception:");

try {

generateException(input);

} catch (Exception e) {

System.out.println("Exception caught in main: " + e);

}

System.out.println("\nHandling Exception:");

handleException(input);

sc.close();

}

}

Write a program to demonstrate ArrayIndexOutOfBoundsException

Hint =>

Define a variable of array of names and take input from the user

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.

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

From the main Firstly call the method to generate the Exception then call the method to handle the RuntimeException

import java.util.Scanner;

public class ArrayIndexDemo {

public static void generateException(String[] names, int index) {

System.out.println("Accessing name at index " + index + ": " + names[index]);

}

public static void handleException(String[] names, int index) {

try {

System.out.println("Accessing name at index " + index + ": " + names[index]);

} catch (ArrayIndexOutOfBoundsException e) {

System.out.println("Caught ArrayIndexOutOfBoundsException: " + e.getMessage());

} catch (RuntimeException e) {

System.out.println("Caught RuntimeException: " + e.getMessage());

}

}

public static void main(String[] args) {

String[] names = {"Manshi", "Garv", "Jahnavi", "Payal", "Vanshika"};

Scanner sc = new Scanner(System.in);

System.out.print("Enter index to access name (0 to " + (names.length - 1) + "): ");

int index = sc.nextInt();

System.out.println("\nGenerating Exception:");

try {

generateException(names, index);

} catch (Exception e) {

System.out.println("Exception caught in main: " + e);

}

System.out.println("\nHandling Exception:");

handleException(names, index);

sc.close();

}

}

Write a program to convert the complete text to uppercase and compare the results

Hint =>

Take user input using the Scanner nextLine() method to take the complete text into a String variable

Write a method using the String built-in charAt() method to convert each character if it is lowercase to the Upper Case. 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

Write a method to compare two strings using the charAt() method and return a boolean result

In the main() use the String built-in method toLowerCase() to get the Uppercase Text and compare the two strings using the user-defined method. And finally display the result

import java.util.Scanner;

public class UpperCaseConversion {

public static String toUpperManual(String text) {

String result = "";

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

char ch = text.charAt(i);

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

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

} else {

result += ch;

}

}

return result;

}

public static boolean compareStrings(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 sc = new Scanner(System.in);

System.out.print("Enter a line of text: ");

String input = sc.nextLine();

String manualUpper = toUpperManual(input);

String builtInUpper = input.toUpperCase();

boolean areEqual = compareStrings(manualUpper, builtInUpper);

System.out.println("Manual Uppercase: " + manualUpper);

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

System.out.println("Are both equal? " + areEqual);

sc.close();

}

}

Write a program to convert the complete text to lowercase and compare the results

Hint =>

Take user input using the Scanner nextLine() method to take the complete text into a String variable

Write a method using the String built-in charAt() method to convert each character if it is lowercase to the Upper Case. 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

Write a method to compare two strings using the charAt() method and return a boolean result

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

import java.util.Scanner;

public class LowerCaseConversion {

public static String toLowerManual(String text) {

String result = "";

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

char ch = text.charAt(i);

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

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

} else {

result += ch;

}

}

return result;

}

public static boolean compareStrings(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 sc = new Scanner(System.in);

System.out.print("Enter a line of text: ");

String input = sc.nextLine();

String manualLower = toLowerManual(input);

String builtInLower = input.toLowerCase();

boolean areEqual = compareStrings(manualLower, builtInLower);

System.out.println("Manual Lowercase: " + manualLower);

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

System.out.println("Are both equal? " + areEqual);

sc.close();

}

}