BYAGARI PAVAN PBL ID: J\_251890123

**Hands-on Assignments for String, StringBuffer**

1. **Write a Program to check whether a given String is Palindrome or not.**

import java.util.Scanner;

public class PalindromeCheck {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

// Input from user System.out.print("Enter a string: "); String str = sc.nextLine();

String cleanStr = str.replaceAll("\\s+", "").toLowerCase();

String reversed = new StringBuilder(cleanStr).reverse().toString();

// Check palindrome

if (cleanStr.equals(reversed)) { System.out.println("'" + str + "' is a Palindrome.");

} else {

System.out.println("'" + str + "' is NOT a Palindrome.");

}

sc.close();

}

}

**Output:**

Enter a string: madam 'madam' is a Palindrome

1. **Write a java program that will concatenate 2 strings and return the result. The result should be in lowercase.**

**Note: If the concatenation creates a double-char, then one of the characters need to be omitted.**

**Example1) i/p: Sachin, Tendulkar o/p:sachin tendulkar**

**Example2) i/p:Mark, kate**

**o/p:markate**

import java.util.Scanner;

public class ConcatenateStrings { public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter first string: "); String str1 = sc.nextLine(); System.out.print("Enter second string: "); String str2 = sc.nextLine();

str1 = str1.toLowerCase(); str2 = str2.toLowerCase(); String result;

if (str1.charAt(str1.length() - 1) == str2.charAt(0)) { result = str1 + str2.substring(1);

} else {

result = str1 + " " + str2; // add space only if not overlapping

}

System.out.println("Result: " + result); sc.close();

}

}

1. **Given a string, return a new string made of 'n' copies of the first 2 chars of the original string**

**where 'n' is the length of the string. Example1)**

**i/p:Wipro**

**o/p: WiWiWiWiWi**

import java.util.Scanner; public class RepeatFirstTwo {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

System.out.print("Enter a string: "); String str = sc.nextLine();

int n = str.length(); // length of original string

String firstTwo = (n >= 2) ? str.substring(0, 2) : str;

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

result.append(firstTwo);

}

System.out.println("Output: " + result.toString()); sc.close();

}

}

1. **Write a java program that will return the first half of the string, if the length of the string is even. It should return null for odd length string.**

**Example1) i/p: TomCat o/p: Tom**

**Example2) i/p: Apron o/p:null**

import java.util.Scanner; public class FirstHalfString {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

System.out.print("Enter a string: "); String str = sc.nextLine();

if (str.length() % 2 == 0) {

String firstHalf = str.substring(0, str.length() / 2);

System.out.println("Output: " + firstHalf);

} else {

System.out.println("Output: null");

}

sc.close();

}

}

1. **Write a java program that accepts a string and returns a new string without the first and last character of the input string.**

**Example1) i/p: Suman**

**/p:uma**

import java.util.Scanner;

public class RemoveFirstLast {

public static void main(String[] args) { Scanner sc = new Scanner(System.in); System.out.print("Enter a string: "); String input = sc.nextLine();

String result = removeFirstLastChar(input); System.out.println("Output: " + result);

sc.close();

}

public static String removeFirstLastChar(String str) { if (str == null || str.length() <= 2) {

return ""; // Return empty string if input is too short

}

return str.substring(1, str.length() - 1);

}

}

1. **Given 2 strings, a and b, return a new string of the form short+long+short, with the shorter string on the outside and the longer string on the inside.**

**The strings will not be the same length, but they may be empty (length 0). If input is "hi" and "hello", then output will be "hihellohi".**

public class StringCombiner {

public static String shortLongShort(String a, String b) {

// Check which string is shorter and concatenate accordingly if (a.length() < b.length()) {

return a + b + a; // Shorter string a

} else {

return b + a + b; // Shorter string b

}

}

public static void main(String[] args) {

// Example usage

String result = shortLongShort("hi", "hello"); System.out.println(result); // Output: hihellohi

// Additional test cases

System.out.println(shortLongShort("abc", "de")); // Output: deabcde System.out.println(shortLongShort("a", "bcdef")); // Output: abcdefa System.out.println(shortLongShort("", "hello")); // Output: hello System.out.println(shortLongShort("hi", "")); // Output: hihi

}

}

1. **Given a string, if the first or last chars are 'x', return the string without those 'x' chars, otherwise return the string unchanged.**

**If the input is "xHix", then output is "Hi".**

**If the input is "America", then the output is "America".**

public class Main {

public static String withoutX(String str) {

if (str.length() > 0 CC str.charAt(0) == 'x') { str = str.substring(1);

}

if (str.length() > 0 CC str.charAt(str.length() - 1) == 'x') { str = str.substring(0, str.length() - 1);

}

return str;

}

public static void main(String[] args) { System.out.println(withoutX("xHix")); // Output: "Hi" System.out.println(withoutX("America")); // Output: "America" System.out.println(withoutX("x")); // Output: "" System.out.println(withoutX("xx")); // Output: ""

System.out.println(withoutX("")); // Output: ""

}

}

1. **Write a Java program that in accepts a string (with it). The program should return a new string in which the following characters are removed-\*, the characters that are to the left and right of \* Example1)**

**i/p:ab\*cd o/p:ad**

import java.util.Scanner;

public class RemoveStarNeighbors {

public static String removeStar(String str) { if (str == null || str.isEmpty()) return str;

StringBuilder sb = new StringBuilder(); int n = str.length();

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

// Skip current character if it is '\*' or adjacent to '\*' if (str.charAt(i) == '\*') continue;

if (i > 0 CC str.charAt(i - 1) == '\*') continue;

if (i < n - 1 CC str.charAt(i + 1) == '\*') continue;

sb.append(str.charAt(i));

}

return sb.toString();

}

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

System.out.print("Enter a string: "); String input = sc.nextLine();

String result = removeStar(input); System.out.println("Output: " + result);

// Test case

System.out.println(removeStar("ab\*cd")); // Output: "ad" System.out.println(removeStar("\*abc\*")); // Output: "c" System.out.println(removeStar("a\*b\*c")); // Output: "" sc.close();

}

}

**G)Given two strings, a and b, print a new string which is made of the following combination-first character of a, the first character of b, second character of a, second character of b and so on. Any characters left, will go to the end of the result.**

**Example1) i/p:Hello, World**

**o/p:HWeolrllod**

import java.util.Scanner; public class InterleaveStrings {

public static String combineStrings(String a, String b) { if (a == null) a = "";

if (b == null) b = "";

StringBuilder sb = new StringBuilder(); int i = 0, j = 0;

int lenA = a.length(), lenB = b.length(); int minLen = Math.min(lenA, lenB);

// Interleave characters from both strings for (int k = 0; k < minLen; k++) {

sb.append(a.charAt(k));

sb.append(b.charAt(k));

}

// Append remaining characters from the longer string if (lenA > lenB) {

sb.append(a.substring(minLen));

} else if (lenB > lenA) { sb.append(b.substring(minLen));

}

return sb.toString();

}

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

System.out.print("Enter first string: "); String str1 = sc.nextLine();

System.out.print("Enter second string: "); String str2 = sc.nextLine();

String result = combineStrings(str1, str2); System.out.println("Output: " + result);

// Test case

System.out.println(combineStrings("Hello", "World")); // Output: HWeolrllod

sc.close();

}

}

**10) Given a string and an integer n, print a new string made of n repetitions of the last n characters of the string.**

**You may assume that n is between and the length of the string, inclusive. Example1)**

**i/p:Wipro, 3**

**o/p: propropro**

import java.util.Scanner;

public class RepeatLastNChars {

public static String repeatLastN(String str, int n) { if (str == null || n <= 0 || n > str.length()) {

return ""; // Handle invalid input

}

String lastN = str.substring(str.length() - n); // Get last n characters StringBuilder sb = new StringBuilder();

for (int i = 0; i < n; i++) { sb.append(lastN);

}

return sb.toString();

}

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

System.out.print("Enter a string: "); String input = sc.nextLine();

System.out.print("Enter an integer n: "); int n = sc.nextInt();

String result = repeatLastN(input, n); System.out.println("Output: " + result);

// Test case

System.out.println(repeatLastN("Wipro", 3)); // Output: propropro

sc.close();

}

}