CSA09 – Programming in Java

Day 3 Assignment Questions

SUBMITTED BY:S.CHARAN SAI

REGISTER NO:192211519

1. Write a program in Java for dynamically changing the color of Text using

Multithreading.

# **PROGRAM:**

**i**mport java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class dynamictextcolor {

private Thread thread;

private int red = 255;

private int green = 0;

private int blue = 0;

private boolean increasing = false;

public DynamicTextColor() {

thread = new Thread(this);

thread.start();

}

public void run() {

while (true) {

if (increasing) {

if (blue < 255) {

blue += 5;

} else {

increasing = false;

green -= 5;

}

} else {

if (green > 0) {

green -= 5;

} else {

increasing = true;

red -= 5;

}

}

repaint();

try {

Thread.sleep(50);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

public void paint(Graphics g) {

super.paint(g);

g.setColor(new Color(red, green, blue));

g.setFont(new Font("Arial", Font.BOLD, 24));

g.drawString("Hello, world!", 50, 50);

}

public static void main(String[] args) {

JFrame frame = new JFrame("Dynamic Text Color");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(300, 200);

frame.add(new DynamicTextColor());

frame.setVisible(true);

}

}

2. Differentiate Multiprocessing and Multithreading. Display Multiplication table for 5

and 10 using various stages of life cycle of the thread by generating a suitable code in

Java.

PROGRAM:

import java.util.Scanner;

class multiplicationtable{

private int n;

public MultiplicationTable(int n) {

this.n = n;

}

public void run() {

System.out.println(Thread.currentThread().getName() + " started");

for (int i = 1; i <= 10; i++) {

System.out.println(n + " x " + i + " = " + (n\*i));

try {

Thread.sleep(100);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

System.out.println(Thread.currentThread().getName() + " ended");

}

}

public class ThreadExample {

public static void main(String[] args) {

Thread thread1 = new Thread(new MultiplicationTable(5));

Thread thread2 = new Thread(new MultiplicationTable(10));

System.out.println("Main thread started");

thread1.start();

thread2.start();

System.out.println("Main thread ended");

}

}

3. An ugly number is a positive integer whose prime factors are limited to 2, 3, and 5.

Given an integer n, return true if n is an ugly number.

Example 1:

Input: n = 6

Output: true

Explanation: 6 = 2 × 3

# **PROGRAM:**

import java.util.Scanner;

public class uglynumber {

public boolean isUgly(int n) {

if(n <= 0) {

return false;

}

while(n % 2 == 0) {

n /= 2;

}

while(n % 3 == 0) {

n /= 3;

}

while(n % 5 == 0) {

n /= 5;

}

return n == 1;

}

}

Example 2:

Input: n = 1

Output: true

Explanation: 1 has no prime factors, therefore all of its prime factors are

limited to 2, 3, and 5.

# **PROGRAM:**

import java.util.Scanner;

public class uglysolution {

public boolean isUgly(int n) {

if (n <= 0) {

return false;

}

while (n % 2 == 0) {

n /= 2;

}

while (n % 3 == 0) {

n /= 3;

}

while (n % 5 == 0) {

n /= 5;

}

return n == 1;

}

}

Example 3:

Input: n = 14

Output: false

Explanation: 14 is not ugly since it includes the prime factor 7.

Constraints:

-231 &lt;= n &lt;= 231 - 1

class Solution {

public:

bool isUgly(int n) {

}

}

# **PROGRAM:**

import java.util.Scanner;

public class ugly {

public boolean isUgly(int n) {

if (n <= 0) {

return false;

}

while (n % 2 == 0) {

n /= 2;

}

while (n % 3 == 0) {

n /= 3;

}

while (n % 5 == 0) {

n /= 5;

}

return n == 1;

}

}

4. The Fibonacci numbers, commonly denoted F(n) form a sequence, called the

Fibonacci sequence, such that each number is the sum of the two preceding ones,

starting from 0 and 1. That is,

F(0) = 0, F(1) = 1

F(n) = F(n - 1) + F(n - 2), for n &gt; 1.

Given n, calculate F(n).

Example 1:

Input: n = 2

Output: 1

Explanation: F(2) = F(1) + F(0) = 1 + 0 = 1.

# **PROGRAM:**

import java.util.Scanner;

public class fibonacci {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter a number to find the corresponding Fibonacci number: ");

int n = input.nextInt();

int fibNum = fibonacci(n);

System.out.println("The Fibonacci number for " + n + " is " + fibNum);

}

public static int fibonacci(int n) {

if (n <= 1) {

return n;

}

return fibonacci(n-1) + fibonacci(n-2);

}

}

Example 2:

Input: n = 3

Output: 2

Explanation: F(3) = F(2) + F(1) = 1 + 1 = 2.

# **PROGRAM:**

import java.util.Scanner;

public class fibonacci2 {

public static void main(String[] args) {

int n = 3;

int fib = fibonacci(n);

System.out.println("Fibonacci number at position " + n + " is " + fib);

}

public static int fibonacci(int n) {

if (n <= 1) {

return n;

}

return fibonacci(n - 1) + fibonacci(n - 2);

}

}

Example 3:

Input: n = 4

Output: 3

Explanation: F(4) = F(3) + F(2) = 2 + 1 = 3.

Constraints:

0 &lt;= n &lt;= 30

class Solution {

public:

int fib(int n) {

}

}

# **PROGRAM:**

import java.util.Scanner;

public class fibonacci3 {

public static void main(String[] args) {

int n = 4;

int fib = fibonacci(n);

System.out.println("Fibonacci number at position " + n + " is " + fib);

}

public static int fibonacci(int n) {

if (n <= 1) {

return n;

}

int fib = 1;

int prevFib = 1;

for (int i = 2; i < n; i++) {

int temp = fib;

fib += prevFib;

prevFib = temp;

}

return fib;

}

}

5. Removing duplicate elements in java : Find/Debug the errors and get output

class duplicate

{

// Function to remove duplicate elements

// This function returns new size of modified

// array.

static int removeDuplicates(int arr[], int n)

{

// Return, if array is empty

// or contains a single element

if (n==0 || n==1)

return n;

int[] temp = new int[n];

// Start traversing elements

int j = 0;

for (int j=0; i&lt;n-1; i++)

// If current element is not equal

// to next element then store that

// current element

if (arr[i] != arr[i+1])

temp[j++] = arr[i];

// Store the last element as whether

// it is unique or repeated, it hasn&#39;t

// stored previously

temp[j++] = arr[n-1];

// Modify original array

for (int i=0; i&lt;j; i++)

arr[i] = temp[i];

return j;

}

public static void main (String[] args)

{

it arr[] = {10, 20, 20, 30, 40, 40, 40, 50, 50};

int n = arr.length;

n = removeDuplicates(arr);

// Print updated array

for (int i=0; i&lt;n; i++)

System.out.print(arr[i]+&quot; &quot;);

}

# **PROGRAM:**

import java.util.\*;

class duplicate {

static int removeDuplicates(int arr[], int n) {

if (n==0 || n==1) return n;

int[] temp = new int[n];

nt j = 0;

for (int i=0; i<n-1; i++) {

temp[j++] = arr[i];

}

}

temp[j++] = arr[n-1]

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

arr[i] = temp[i];

}

return j;

}

public static void main(String[] args) {

int arr[] = {10, 20, 20, 30, 40, 40, 40, 50, 50};

int n = arr.length;

n = removeDuplicates(arr, n);

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

System.out.print(arr[i] + " ");

}

}

}

6. Write a program to reverse a word using loop? (Not to use inbuilt functions)

Sample Input:

String: TEMPLE

Sample Output:

Reverse String: ELPMET

# **PROGRAM:**

import java.util.Scanner;

public class reverseword {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String word = scanner.nextLine();

String reversed = "";

for(int i = word.length() - 1; i >= 0; i--) {

reversed += word.charAt(i);

}

System.out.println("Reverse String: " + reversed);

scanner.close();

}

}

7. Write a program to convent the given string to integer?

Sample Input:

String: 1234

Sample Output:

Out put String: 1234

# **PROGRAM:**

import java.util.Scanner;

public class stringtoint {

public static void main(String[] args) {

String inputString = "1234";

int outputInt = Integer.parseInt(inputString);

System.out.println("Output Integer: " + outputInt);

}

}

8. Write a program to check the entered user name is valid or not. Get both the inputs from

the user.

# **PROGRAM:**

import java.util.Scanner;

public class usernamevalidator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String username = scanner.nextLine();

if (isValidUsername(username)) {

System.out.println("Valid username.");

} else {

System.out.println("Invalid username.");

}

}

public static boolean isValidUsername(String username) {

String pattern = "^[a-zA-Z][a-zA-Z0-9\_-]{2,15}$";

return username.matches(pattern);

}

}

9. Write a program that would sort a list of names in alphabetical order Ascending or

Descending, choice get from the user?

Sample Input:

Banana

Carrot

Radish

Apple

Jack

Order(A/D) : A

Sample Output:

Apple

Banana

Carrot

Jack

Radish

# **PROGRAM:**

import java.util.\*;

public class namesorter {

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

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

System.out.println("Enter names (one per line), then press enter twice:");

String name = scanner.nextLine();

while (!name.isEmpty()) {

names.add(name);

name = scanner.nextLine();

}

System.out.println("Sort order (A for ascending, D for descending):");

String sortOrder = scanner.nextLine();

if (sortOrder.equalsIgnoreCase("A")) {

Collections.sort(names);

} else if (sortOrder.equalsIgnoreCase("D")) {

Collections.sort(names, Collections.reverseOrder());

} else {

System.out.println("Invalid sort order specified.");

System.exit(1);

}

System.out.println("Sorted names:");

for (String n : names) {

System.out.println(n);

}

}

}

10. Write a program to print the special characters separately and print number of Special

characters in the line?

**PROGRAM:**:

import java.util.Scanner;

public class specialcharacters {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String input = scanner.nextLine();

int specialCount = 0;

System.out.print("Special characters: ");

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

char c = input.charAt(i);

if (!Character.isLetterOrDigit(c) && !Character.isWhitespace(c)) {

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

specialCount++;

}

}

System.out.println("\nNumber of special characters: " + specialCount);

}

}

11. Write a program to print the number of vowels in the given statement?

Sample Input:

Saveetha School of Engineering

Sample Output:

Number o vowels = 12

# **PROGRAM:**

import java.util.Scanner;

public class countvowels {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String str = scanner.nextLine().toLowerCase();

int count = 0;

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

char ch = str.charAt(i);

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

count++;

}

}

System.out.println("Number of vowels: " + count);

}

}

12. Write a program to print consonants and vowels separately in the given word

Sample Input:

Given Word: Engineering

Sample Output:

Consonants: n g n r n g

Vowels: e i e ei

# **PROGRAM:**

import java.util.Scanner;

public class separateconsonantsandvowels {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String word = scanner.nextLine();

scanner.close();

String consonants = "";

String vowels = "";

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

char c = word.charAt(i);

if (Character.isLetter(c)) {

if (isVowel(c)) {

vowels += c;

} else {

consonants += c;

}

}

}

System.out.println("Consonants: " + consonants);

System.out.println("Vowels: " + vowels);

}

private static boolean isVowel(char c) {

return "aeiouAEIOU".indexOf(c) != -1;

}

}

13. Write a program that finds whether a given character is present in a string or not. In case

it is present it prints the index at which it is present. Do not use built-in find functions to

search the character.

Sample Input:

Enter the string: I am a programmer

Enter the character to be searched: p

Sample Output:

P is found in string at index: 8

Note: Check for non available Character in the given statement as Hidden Test case.

# **PROGRAM:**

import java.util.Scanner;

public class charactersearch {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String inputString = scanner.nextLine();

System.out.print("Enter the character to be searched: ");

char searchChar = scanner.next().charAt(0);

int index = -1;

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

if (inputString.charAt(i) == searchChar) {

index = i;

break;

}

}

if (index == -1) {

System.out.println(searchChar + " is not found in string.");

} else {

System.out.println(searchChar + " is found in string at index: " + index);

}

}

}

14. Write a program to arrange the letters of the word alphabetically in reverse order

Sample Input:

Enter the word: MOSQUE

Sample Output:

Alphabetical Order: U S Q O M E

Test Case:

1. HYPOTHECATION

2. MATRICULATION

3. MANIPULATION

# **PROGRAM:**

import java.util.Arrays;

import java.util.Scanner;

public class reversealphabeticalorder {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the word: ");

String word = scanner.nextLine();

char[] chars = word.toCharArray();

Arrays.sort(chars);

System.out.print("Alphabetical Order: ");

for (int i = chars.length - 1; i >= 0; i--) {

System.out.print(chars[i] + " ");

}

}

}

15. Write a program that accepts a string from user and displays the same string after

removing vowels from it.

Sample Input &amp; Output:

Enter a string: we can play the game

The string without vowels is: w cn ply thgm

# **PROGRAM:**

import java.util.Scanner;

public class removevowels {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

String str = input.nextLine();

String result = removeVowels(str);

System.out.println("The string without vowels is: " + result);

}

public static String removeVowels(String str) {

String vowels = "aeiouAEIOU";

StringBuilder result = new StringBuilder();

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

char ch = str.charAt(i);

if (vowels.indexOf(ch) == -1) {

result.append(ch);

}

}

return result.toString();

}

}