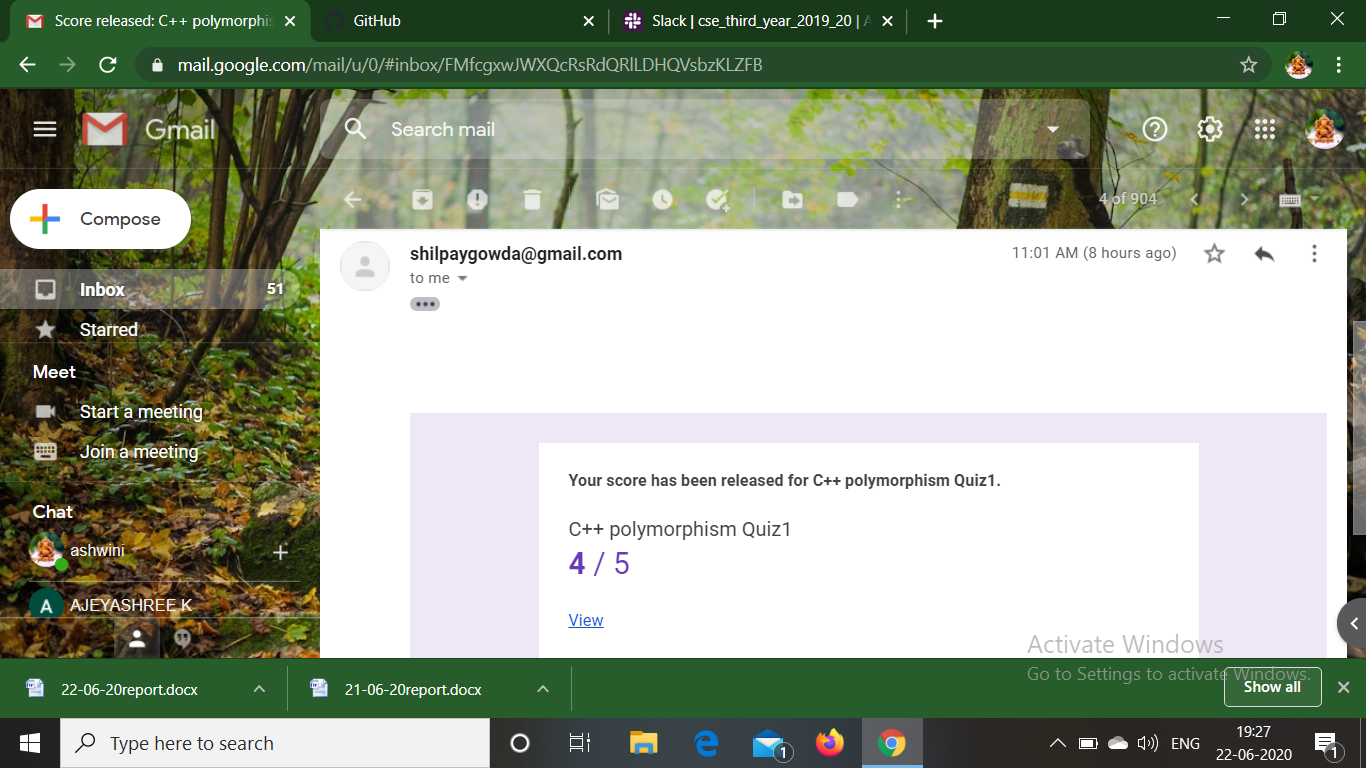
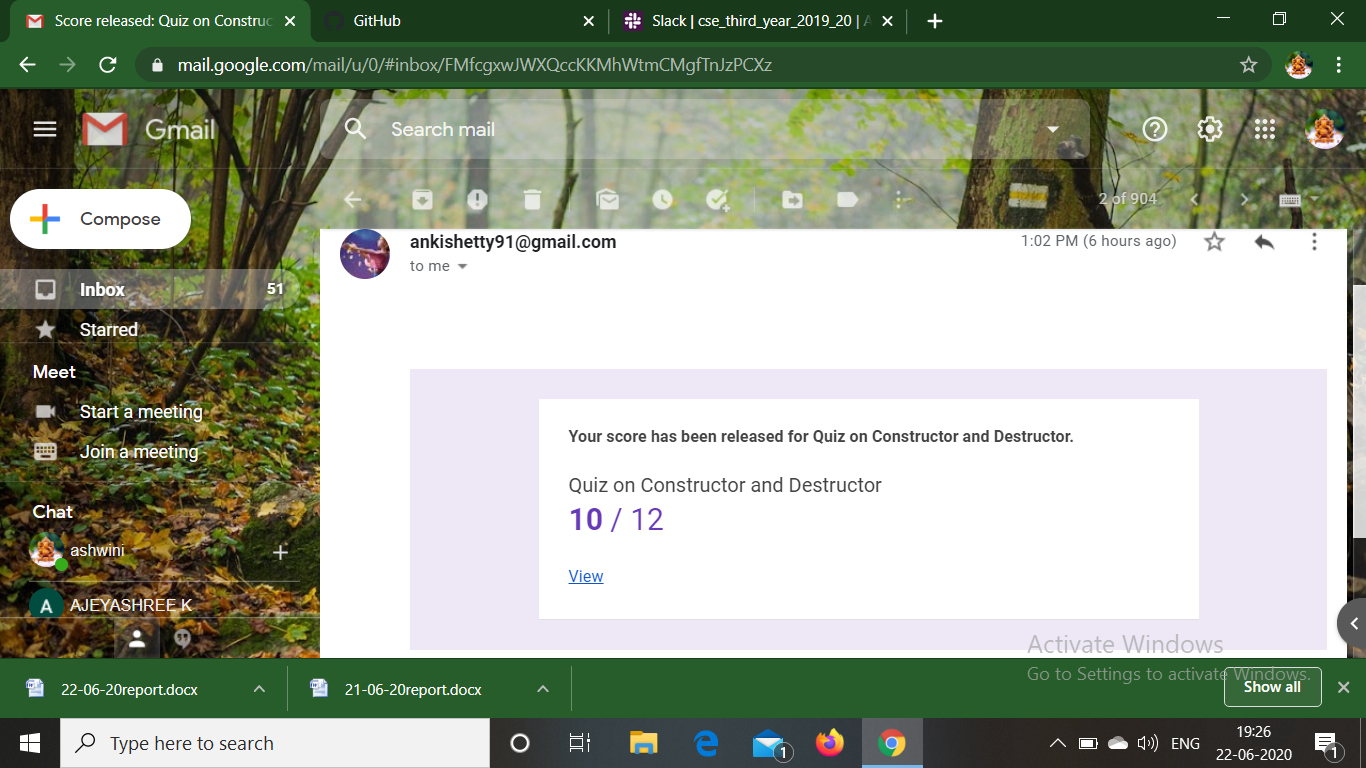
**DAILY ONLINE ACTIVITIES SUMMARY**

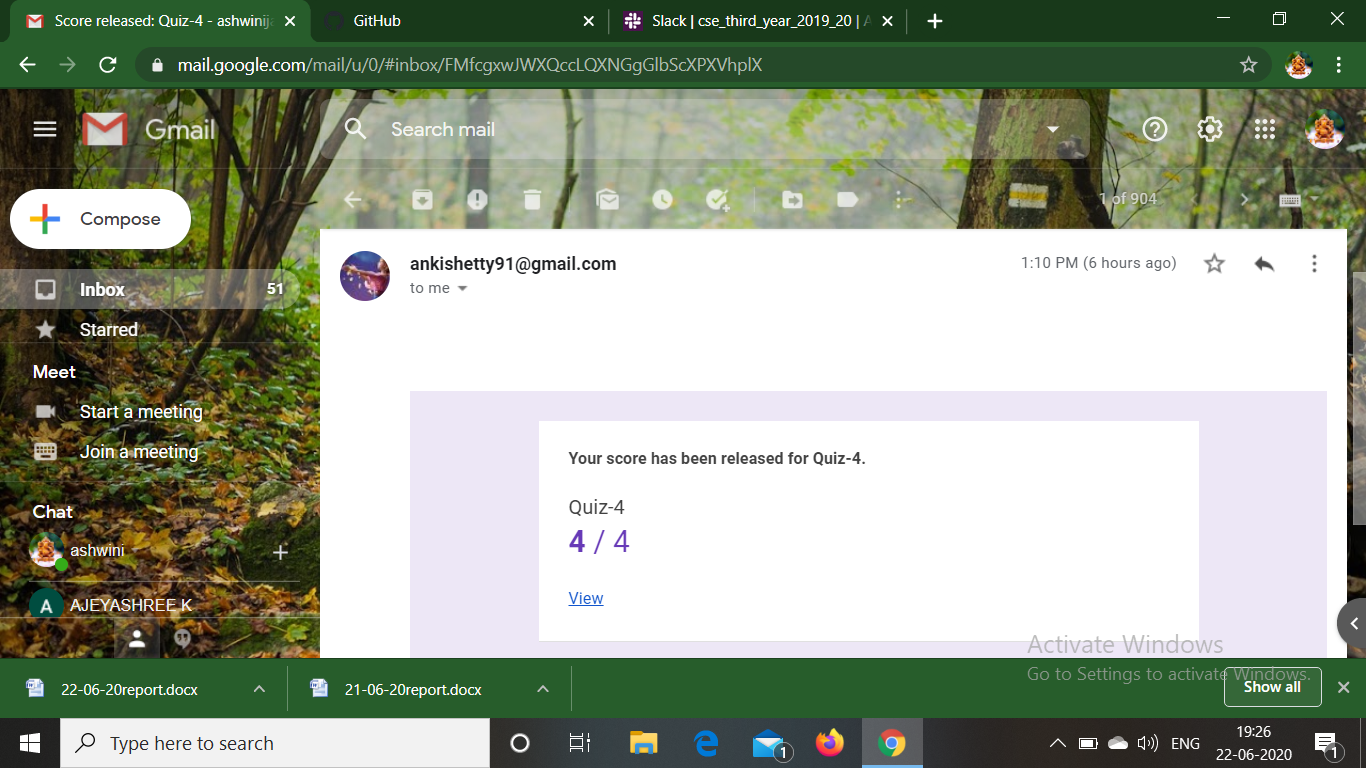
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **22-06-2020** | | | | | **Name:** | | **Ashwini S Jadamali** | |
| **Sem & Sec** | **6th  Sem ‘A’ Sec** | | | | | **USN:** | | **4AL17CS018** | |
| **Online Test Summary** | | | | | | | | | |
| **Subject** | | **Programming in c++ .** | | | | | | | |
| **Max. Marks** | | **C ++quiz1=5**  **C++quiz2=5**  **C++quz3=12**  **C++quiz4=4** | | **Score** | | | | **Quiz1=not evaluated**  **Quiz2=4**  **Quiz4=10**  **Quiz4=4** | |
| **Pre-Placement Training Summary** | | | | | | | | | |
| **Course** | **Workshop of C++ programming.** | | | | | | | | |
| **Faculty** | | | **Ankitha mam.**  **Shilpa mam.** | | | | **Duration** | | **4 hours** |
| **Coding Challenges** | | | | | | | | | |
| **Problem Statement:1.** Write a Java program to find maximum repeated words from a file. | | | | | | | | | |
| **Status: done** | | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | | |
| **If yes Repository name** | | | | | Daily Report =https://github.com/ashwinijadamali/online-coding-activites | | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | | |

**Class and Quiz Snapshots:**

**Programming in C++:**

****

****

****

**Coding Challenge:**

**1.** Write a Java program to find maximum repeated words from a file.

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

import java.util.HashMap;

import java.util.Map.Entry;

import java.util.Set;

public class RepeatedWordInFile

{

public static void main(String[] args)

{

HashMap<String, Integer> wordCountMap = new HashMap<String, Integer>();

BufferedReader reader = null;

try

{

reader = new BufferedReader(new FileReader("C:\sample.txt"));

String currentLine = reader.readLine();

while (currentLine != null)

{

String[] words = currentLine.toLowerCase().split(" ");

for (String word : words)

{

if(wordCountMap.containsKey(word))

{

wordCountMap.put(word, wordCountMap.get(word)+1);

}

else

{

wordCountMap.put(word, 1);

}

}

currentLine = reader.readLine();

}

String mostRepeatedWord = null;

int count = 0;

Set<Entry<String, Integer>> entrySet = wordCountMap.entrySet();

for (Entry<String, Integer> entry : entrySet)

{

if(entry.getValue() > count)

{

mostRepeatedWord = entry.getKey();

count = entry.getValue();

}

}

System.out.println("The most repeated word in input file is : "+mostRepeatedWord);

System.out.println("Number Of Occurrences : "+count);

}

catch (IOException e)

{

e.printStackTrace();

}

finally

{

try

{

reader.close();

}

catch (IOException e)

{

e.printStackTrace();

}

}

}

}