Object-Oriented-Programming Coursework 4 Report

20191339 chanhwiHwang

In this coursework, I implemented two programs, the first program reads “Input.txt” file, which contains the information of the number of cases, number of students and each student’s scores, calculate the ratio of the student who has less score than average at each cases, and write the information to “Output.txt” as external file. And the second one is to count the number of a word contained in “Paragraph.txt” and then print it without distinguishing the lowercase and uppercase characters. So, in first program, I defined two file streams, one is an input stream to read the input values from “Input.txt” and the other is an output stream to write the desired output to “Output.txt” file. Then, using input stream, first read the number of cases and used for-loops from 1 to the number of cases to use the iterate index variable(such as I) as the index of cases in writing file. Then, I defined sum(int), num(int) and integer array, nums with “num” length, where num came from input stream. Then, I used second for-loop to find the sum of scores and store each score to nums by repeatedly reading values from input stream. After the second loop, I calculated the average score like sum / num and used loop again with same level as previous one to find the number of students who has lower score than average. After that, I wrote the number of students with lower score than average / num * 100 by using output stream with proper format at each case. At the end, after use of file streams, I closed them. Second program was implemented with similar processes. First, receive a word from user as input and by using input stream, open the “Paragraph.txt” file. Using while loop, unless open stream’s “fail()” function returns true, which means it reaches at the end of the file, read from input stream and first compare the length of received word with read string. Then, if the length is same, repeatedly compare each character of them using “tolower()” function to compare them

without considering upper or lower case. To distinguish the different(there is character that is different) case with same(all characters are same) case, I defined “isSame” boolean variable, which is initially true. Then, when a character is different while comparing, by if statement, it meets break with assignment statement that isSame = false. So, only the case with not meeting break – all characters of them are same – is counted as same word with received word. After the while loop, it prints the number of counted words.