1811ICT/2807ICT/7001ICT Programming Principles Workshop 7

School of Information and Communication Technology

Griffith University

|  |  |
| --- | --- |
| *Goals* | This workshop focusses on everything in the course up to files. |
| *When* | Week 8 |

# Before your workshop class:

* Read the whole document.
* Review the lecture notes of sections 1 to 20.
* **Complete the pre-workshop questions posted on the course website**.

# Workshop activities

At any stage, when you are stuck, *ask your workshop instructor*!

## Problem 1

*Problem:* Write a program that prompts the user for the name of a source file to read and the name of the target file to write and copy the content of the source file to, but with all empty lines removed. It then outputs the number of empty lines that have been removed as indicated below.

Source file name: string\_doc.txt

Target file name: string\_doc\_nonempty.txt

Lines removed: 16

*Testing*: Test your code with the file *P1\_v1.txt*. Read from this file and choose your own file name for the target file (file that you will be writing to).



Did you check what happens if the source file does not exist? How can you deal with this error?

## Problem 2

*Problem:* Write a program that prompts for the user for the name of a file, and then prints the first two lines and the last two lines of the file.

File name: yesterday.txt

Yesterday Once More

When I was young

Still sound so good to me

As they melt the years away

*Testing*: Test your code with the files *yesterday.txt* (compare it to the output above), *P2\_v1.txt* and *P2\_v2.txt*.



If you would use pre- and postconditions for error checking, what would you include in your code?

## Problem 3

*Problem:* Write a program that prompts the user for the name of a file containing numbers in each line and prints the average of each line. Assume each line contains numbers only and they are separated by spaces.

File name: scores.txt

Average of line 1: 60.0

Average of line 2: 91.75

Average of line 3: 48.75

Average of line 4: 56.25

*Testing*: Test your code with *scores.txt* and compare your output with what is indicated above. Then test it with *P3\_v1.txt* and *P3\_v2.txt*.

How would you extend the code to also calculate and print out the smallest average, the largest average and the average of all the lines.



## Problem 4

*Problem:* The Unix tool *wc* counts the numbers of characters, words and lines in a file. Write your own version of *wc* that prompts the user for the name of the file to read, and then prints the counts. Assume a word may contain letters, digits, symbols and a mixture of these, but no spaces. Hyphenated words, e.g. large-scale, shall be considered as one word.

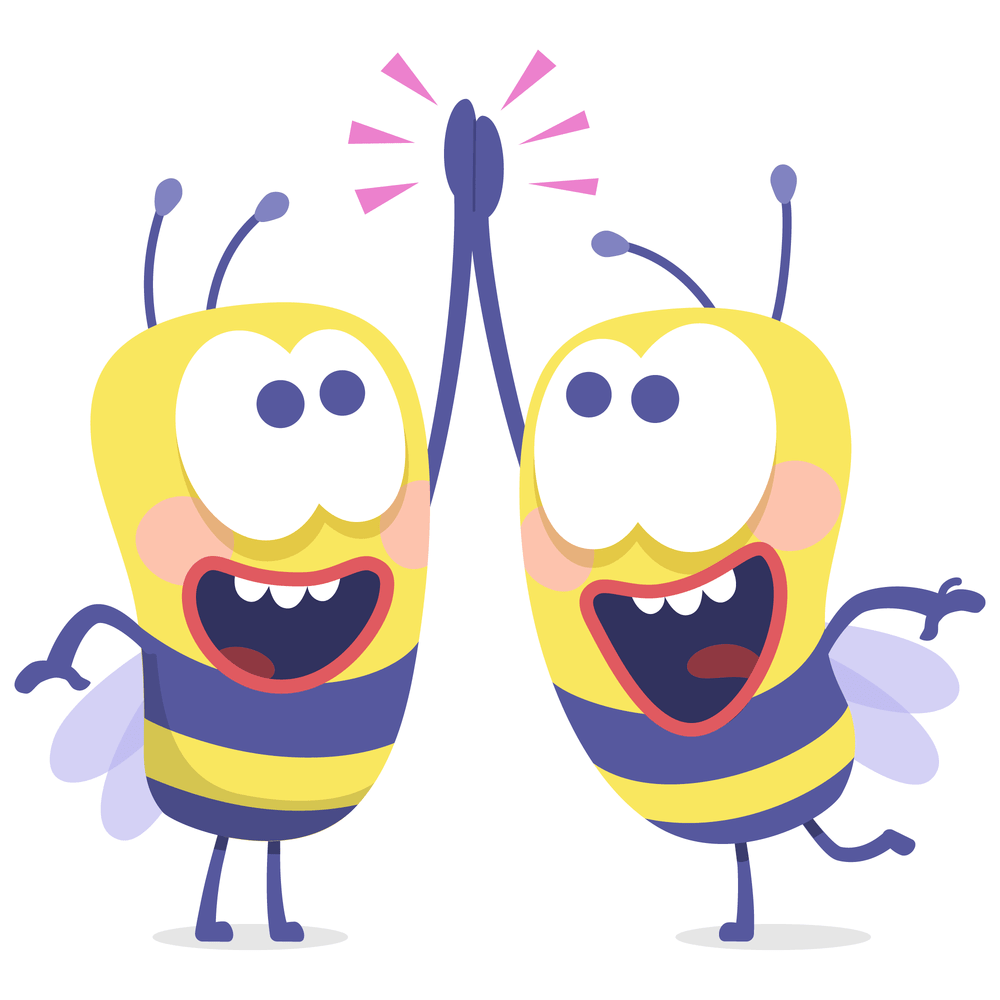
File name: python.txt

Characters: 1227

Words: 176

Lines: 10

*Testing*: Test your code with *python.txt* and compare your output to what is indicated above. Then test your code with *P4\_v1.txt* and *P4\_v2.txt*.



Well done for finishing these activities!