

# COMP2045 Coursework 1

## Introduction

This coursework is worth **5% of the module mark**. It requires you to write a C or C++ program that solves the task described below. The deadline for this exercise is **23:55pm on Tuesday 5th March 2019**.

Read the entire document before beginning the exercise.

If you have any questions about this exercise, please ask in the Q&A forum on Moodle, after a lecture, in a lab, or during the advertised office hours. Do not post your program or parts of your program to Moodle as you are not allowed to share your coursework programs with other students. If any questions require this exercise to be clarified then this document will be updated and everyone will be notified via Moodle.

## Submission

You must submit a single C or C++ source code file containing all your code for this exercise. The file must be called [search.c](#) (or [search.cpp](#) if it is C++ code) and it should contain all the source code.

The first line of the file should be a comment which contains your student ID number, username, and full name, of the form:

```
// 6512345 zy12345 Joe Blogs
```

If the file is in C, it must compile without warnings or errors when I use the command

```
gcc -std=c99 -lm -Wall -Wformat -Wwrite-strings search.c -o search
```

If the file is in C++, it must compile without warnings or errors when I use the command

```
g++ -std=c++11 search.cpp -o search
```

These commands will be run on our Linux server cslinux. If they do not compile, for any reason, then you will lose all the marks for testing. The completed source code file should be uploaded to the Coursework 1 Submission link on the Moodle page.

Late submissions: Late submissions will lose 5 percentage points per hour, rounded up to the next whole hour. No late submissions will be accepted more than 24 hours after the exercise deadline. If you have extenuating circumstances you must file them before the deadline.

## **Task**

Your task is to write a C or C++ program that will check whether a string is contained in a text file. The filename of the text file will be specified as a [command line argument](#). Your program should load this file and then prompt the user to enter a string to search for. There should not be limitation on the length of input string.

After the user has entered a string, the program should output how many occurrence of the string is contained in the file. It should repeatedly do this until the user enters an empty string.

If the program is run without the correct number of command line parameters, it should exit with the message "Invalid command line arguments. Usage: ./search textfile". If the file cannot be opened, print the error message "Cannot open file".

## **Example input/output**

Given the following text file (assume that the filename is helloworld.txt, with the file in the current directory)

```
Hello World!  
Hello C++!  
Hello UNNC!
```

Running the program and just pressing return:

```
zlizpd3 $ ./search
Invalid command line arguments. Usage: ./search textfile
zlizpd3 $
```

Running the program:

```
zlizpd3 $ ./search helloworld.txt
Search for:
zlizpd3 $
```

Running the program:

```
zlizpd3 $ ./search helloworld.txt
Search for: Hello
3 occurrence found in the file
Search for: hello
0 occurrence found in the file
Search for: Hello C++
1 occurrence found in the file
Search for: e
3 occurrence found in the file
Search for:
zlizpd3 $
```

Running the program with a wrong filename:

```
zlizpd3 $ ./search missing.txt
Cannot open file
zlizpd3 $
```

## **Marking**

Your program should correctly implement the task requirements. A number of tests will be run against your program with different input data designed to test if this is the case for each individual requirement. The tests themselves are secret but general examples of the tests might be:

- Does the program work with the example I/O in the question?
- Does the program correctly deal with different input?
- Does the program output match the required format?

If your program does not compile then you will lose all marks. Each warning message leads to -1 mark.

## **Plagiarism**

**You should complete this coursework on your own. Anyone suspected of plagiarism will be investigated and punished in accordance with the university policy on plagiarism (see your student handbook and the University Quality Manual). This may include a mark of zero for this coursework.**

**You should write the source code required for this assignment yourself. If you use code from other sources (books, web pages, etc), you should use comments to acknowledge this. You must not copy or share source code with other students. You must not work together on your solution. You can informally talk about higher-level ideas but not to a level of detail that would allow you all to create the same source code.**

**Remember, it is quite easy for experienced lecturers to spot plagiarism in source code. We also have automated tools that can help us identify shared code, even with modifications designed to hide copying. If you are having problems you should ask questions rather than plagiarize. If you are not able to complete the exercise then you should still submit your incomplete program as that will still get you some of the marks for the parts you have done (but make sure your incomplete solution compiles and partially runs!).**

**If I have concerns about a submission, I may ask you to come to my office and explain your work in your own words.**