

Preparation for COMP519 Exercises, Labs, and Assignments

1 Introduction

The module COMP519 Web Programming normally uses practical sessions to allow students to explore elements of the various languages that we cover on the module, guided by worksheets with exercises that students complete. These practical sessions would take place in the computer labs of the Department of Computer Science. During a session students are supported by module co-ordinator and demonstrators. Whenever a student has a problem with an exercise they could ask a demonstrator for help. Student and demonstrator would look at the problem together and try to find a solution. This sometimes requires the demonstrator to ‘take control’ of the student’s Windows PC.

Another aspect of COMP519 is that almost all programming takes place on the departmental Linux systems, not the Windows PCs in the labs. The Windows PCs are merely used to access those Linux systems remotely. The preferred software for doing so is MobaXterm.

The challenge in 2020-21 is how to support students while they complete programming exercises. Even if we were to use the departmental computer labs, a demonstrator could not walk up to a student’s PC, talk to the student face-to-face, and possibly ‘take control’ of the student’s PC. This would bring the demonstrator too close to the student and touching the student’s PC also creates a risk.

So, it is clear that a demonstrator can only interact with students at a distance. On the PCs in the departmental computer labs, NetSupport School makes this possible. It allows a demonstrator from a dedicated ‘Management PC’ to monitor what students are doing on the other PCs in the lab, allows them to remotely talk to the students (provided the PCs have headsets) and to take control of a student’s PC. Students can use the NetSupport software on their PC to ask a demonstrator for help. The two main problems with this approach are: (a) the headset are a risk if they are used by more than one student and (b) students who cannot come to the departmental computer labs require a way to connect remotely to the lab PCs.

To solve the second problem, the University has set up the so-called Remote Teaching Centre Service at

<https://www.liverpool.ac.uk/csd/pc-centres/remote/>

However, students using this service are in a strange situation. From their laptop/PC at home, they first connect to a PC in the departmental computer labs, and then connect to the departmental Linux systems. Instead, they could directly connect the departmental Linux system.

A possibility to support students who directly connect from their laptop/PC at home to a departmental Linux system is to use video conferencing software. Such software does not only allow demonstrator and student to talk to each other, but some video conferencing programs also allow to share screens and to grant control to another user. Zoom seems to be a program which works particularly well in that regard.

Your task in the following is to install on your laptop/PC at home the software that we might need for the module.

2 Required Software

2.1 Zoom

From

<https://zoom.us/download>

download a Zoom client for your laptop/PC and install it. Following the instructions at

<https://news.liverpool.ac.uk/2020/05/04/csd-adds-zoom-to-suite-of-video-conferencing-tools/>

log into Zoom: You should use your MWS username (in the format `username@liverpool.ac.uk`) and password. If you encounter problems with that, contact the University's helpdesk via <https://servicedesk.liverpool.ac.uk/>.

2.2 MobaXterm

If you have a Windows PC, then from

<https://mobaxterm.mobatek.net/download.html>

download the free version of MobaXterm and install it. Right now the University has probably not set up your account for your Linux servers, so there is no point in using MobaXterm yet.

If you have a Linux or MacOS PC, then you will use the `ssh` command or find a program similar to MobaXterm for your operating system.

2.3 Remote Teaching Centre Service

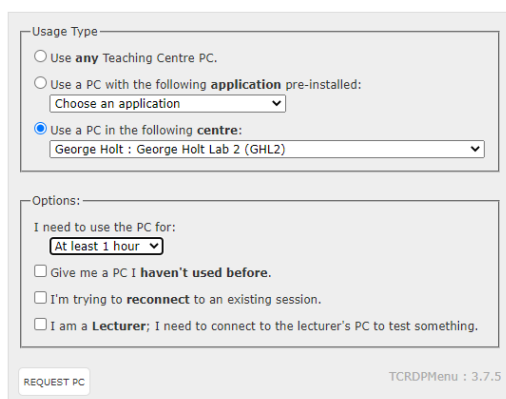
In a web browser visit

<https://www.liverpool.ac.uk/csd/pc-centres/remote/>

Following the instruction in

https://cgi.csc.liv.ac.uk/~ullrich/COMP519/RTCS_User_Guide.pdf

connect to a PC in 'George Holt : George Holt Lab 3 (GHL3)' (see Figure 1). You may have to install the Duo Mobile app for two-factor authentication on your smartphone.




(a) Requesting a PC

Connect to Station

A Teaching Centre station has been reserved for you. Please use the link below to connect to the station.

You must connect to it by the time specified or the reservation will be cancelled to free it up for other users.

| PC | Link | Connect By | Available For |
|----------|---|---------------------|---------------|
| GHL2-059 |  | 2020.10.09 23:39:55 | Over 4 hours |

(b) Connecting to a PC

Figure 1: Remote Teaching Centre Service

3 Organising Your Filestore

Once you are connected to a departmental PC you should explore its filestore.

One of the ways that the systems at the University may differ from your home computer, is that your personal files are not stored on the PC you are using but instead are stored on a central server, and shared across all PCs on campus. This means that whichever PC on campus you log into, you should see the same files.

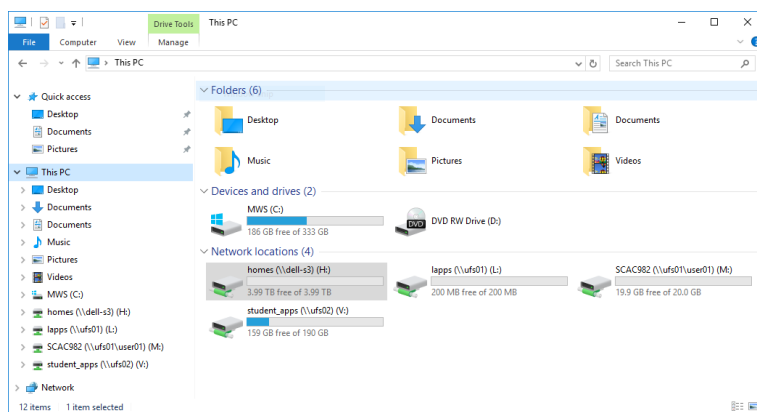
To see what files you currently have access to right click on the Start button, then select the entry “File Explorer” in the menu that opens. The `file explorer` is an example of a type of application called a *file manager*. It displays the following *drives* (Figure 2a), among others:

1. **System (C:)** — the primary partition of the first physical hard disk drive, contains the *operating system*
2. **SGXYZ (M:)** — your University/MWS filestore
(where *sgxyz* is your University account name)

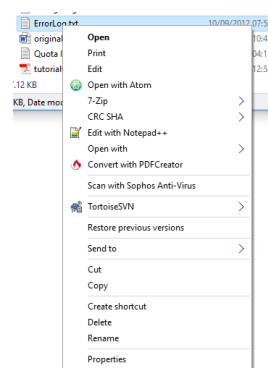
Your personal files should be stored on the **M:** drive, your personal Windows filestore. They will then be available to you on all University PCs. But files on the **M:** drive will not be available on our Linux, Web, and MySQL servers. Instead they must be explicitly transferred to the separate personal filestore you have on these servers.

It is a good idea to organise your filestore in a logical manner. Students typically create a directory for each of their modules, e.g. there would be directories `comp517`, `comp518`, etc. Note that while operating systems nowadays typically allow the use of whitespaces in directory or file names, these are cumbersome when using command line tools; the use of underscores ‘`_`’ instead of whitespaces or the use of CamelCase is preferable. Note also that uppercase and lowercase letters are different on most operating systems, e.g., `comp519` is different to `COMP519`. Use the file manager now to create a directory structure on the **M:** drive that you feel comfortable with.

Now open the top directory of the **M:** drive in the the file manager. That is your *departmental home directory*. Create a file called `www.txt` in it: Right click in the file manager, use “New→Text Document” to create a document, rename it from `New Text Document.txt` to `www.txt`. Move the file `www.txt` to your `comp519` directory and rename the file to `comp519.txt`. Also, create a subdirectory `labs` in the `comp519` directory.



(a) File Explorer



(b) File menu

Figure 2: Dealing with files

```

// HELLO WORLD PROGRAM
// Frans Coenen
// Monday 15 January 1999
// The University of Liverpool, UK

class HelloWorld {
    // ----- METHODS -----

    /* Main method */

    public static void main(String argv[]) {
        System.out.println("Hello World");
    }
}

```

Figure 3: A Java program

4 Editing Text Files

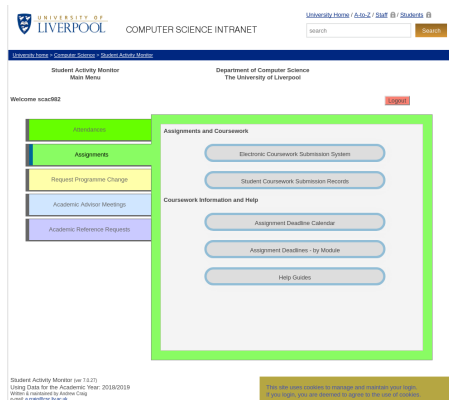
You will regularly have to develop computer programs in a variety of programming languages. While there are *Integrated Development Environments* that make program development easier, on most modules you will just use a *text editor* to write your code. Note that word processors like Microsoft Word or Wordpad are *not* suitable for writing computer programs.

Regarding *text editors*, we will later solely rely on the built-in text editor of MobaXterm. But for now we just want to add a few lines of text to the file `www.txt` and create a file `HelloWorld.java` with some Java code. For that we can use Notepad++, Atom, or Emacs. The latter two have the advantage that they are also available under Linux. Try experimenting with these editors to see which one you prefer to use.

In a web browser locate the reading list for the module COMP519. Then use a text editor to enter the first entry of that reading list into the file `www.txt`.

Also, create a new file `HelloWorld.java` in your `comp519` directory, open it in a text editor, and copy the text shown in Figure 3 from the PDF previewer into the text editor.

Note how different parts of the program get automatically shown in different colours. This is useful, for instance, to spot typos while typing Java keywords. This behaviour of a text editor is called *syntax highlighting*.



(a) Main Menu



(b) Electronic Submission

Figure 4: Student Activity Monitor

5 Submitting Work

Most of the assessed work that you need to assignments and projects needs to be submitted via SAM, the Student Activity Monitor system.

In your web browser, open the URL

<https://sam.csc.liv.ac.uk/COMP/>

and log in using your University account name and password. You see the main menu of the system (Figure 4a). Click on ‘Assignments’ and then on ‘Electronic Coursework Submission System’. You should then be able to select the particular assignment that you want to submit work for, select a file or files to submit and confirm that you have understood and followed the academic integrity rules (Figure 4a). For now, select ‘COMP519-99: Submission Practice’ as the assignment and submit the file `HelloWorld.java` that you have just created. Check that you received a confirmation email once you have done so.