

**CSCU9A2****Spring 2017****Programming Practical 1B:****26 January****Using the Windows 7 command prompt**

Remember to register your practical attendance at the start of each session:

- Double click on the My Computer icon, then on Groups on Wide (the V: drive), then on CSCU9A2, and double click on the Register icon (it should be a camera).

If you get stuck or need help at any time during the practical, ask a demonstrator.

Remember that these are teaching sessions, and during the practicals you should not be surfing the Web, nor texting nor chatting online with your friends.

**Remember: This week's checkpoints must be completed by the end of week 4.**

---


**THIS WORKSHEET:**

*This worksheet introduces you to using Windows 7 at the command prompt, and gives you some Java exercises to be carried mainly at the command prompt.*

---

**BASIC USE OF THE COMMAND PROMPT**

First, launching a command prompt window, customizing its appearance and then some basic navigation and file manipulation at the command prompt:

- *Double-clicking with your mouse in the usual way:* Open **My Computer**, then **C:\**, then **Users**, then the folder labelled with your username. This shows your “profile folder” on *this* PC in an explorer window, which on University lab PCs is not very useful as it does not “move around” from PC to PC with you. On your personal PC it *is* useful – it contains your documents, your desktop, your browser favourites, etc. Just look at it for the moment: a collection of folders and files.
- Click *once* in the blank area towards the right of the current location address bar at the top of the window: it will change from a “breadcrumbs view” to a full “pathname” location for the current folder, for example: **C:\Users\xyz00001** (you can recover the breadcrumbs view by typing the **Esc** key).
- Move the explorer window to one side of the screen.
- Click the Windows **Start** icon , and type **cmd** into the Search box. Windows should find **cmd.exe** and show its name (although the **.exe** extension might be hidden). Press **Enter** or click on **cmd.exe**. A command prompt window will appear.
- Click on the icon at the top left of the command prompt window. Choose Properties from the menu that pops up. This dialogue box allows you to customize the command window – try out some of the customizations.
- When you are working at the command prompt, the system always has a specific “working directory” or “current folder” that will be the default focus of any actions taken – analogous to viewing a specific folder in a desktop explorer window. By default, command windows display the current working directory “pathname” followed by the command prompt symbol **>** whenever the system is waiting for you to type a command. The window that has opened should show something like **C:\Users\xyz00001>**
- Type the command **dir** (in any mixture of upper and lower case, Windows does not care) then **Enter**. The directory contents will be listed on the screen: the name of each item in the working directory, with various additional details. Additional details include the file

size for files, and the indicator <DIR> for subfolders (subdirectories). Look carefully: the listed items *should be exactly the same ones as shown in the explorer window that you opened earlier – if not, please ask a demonstrator!* There will be two unusual items listed: `.` means “this directory”, and `..` means “the enclosing” or “parent” directory.

- The **dir** command has many alternative options: Two ways to find out what is available: Type **help dir**, or type **dir /?** – these work for many commands. **help** on its own lists all the built in commands. If you want more help on any command, just try Googling!
- Try some options: For example **dir /w** gives a directory listing with just folder/file names (folder names enclosed in [ ]). **dir /ah** shows normally hidden files.
- Try **dir /s** This could give a dramatic effect: it lists all files, subfolders, the contents of subfolders, subsubfolders, etc. If it appears to be going wild, with large numbers of folders and files flashing up the screen, then don't panic, it's just obeying your command. (If it does not go wild, then there is a step later where it certainly will!). However, you can tame the wildness in various ways, try them:
  - Hold down the **Ctrl** key and press **C** – this should interrupt the process immediately and Cancel its operation.
  - Or type **Ctrl S** to pause the output, then **Ctrl S** to resume, etc.
  - Or, when you know that this may happen, add `| more` to the command:  
**dir /s | more** (the output from **dir** is “piped” into **more**, which “paginates” the output). The output will stop after every screenful; you type **Space** for the next screen, **Enter** to see one more line, **q** to quit (or **Ctrl C**).  
[The `|` is often on the same key as `\` in the lower left corner of the keyboard. It may look like a vertical bar with a small gap in the middle!]
- Another useful trick is “output redirection” – this allows you to specify that the output from a command is sent to a file instead of to the screen. Try **dir >files.txt** This will create a new file in the current folder – it should automatically appear in the explorer window on your desktop. Type **dir** again and you will see the new file listed.
- Here are three ways to view the contents of a file, try them:
  - The command **type files.txt** will display the contents of the file in the command window.
  - The command **notepad files.txt** will launch NotePad to view/edit the file.
  - The command **start files.txt** will launch the appropriate application to view/edit a **.txt** file (usually NotePad).
- NotePad was easy to launch because the executable application is in the file **notepad.exe** which is in the folder **C:\Windows\System32**, which is mentioned in the system's “path” – a semicolon separated list of folder locations. To see the path list, type **path** at the command prompt.
- The University lab PCs also have a better text editor called TextPad installed (not a Microsoft product). This is not locatable by using the path list (the system admins have not bothered putting its installation folder in the path). However, any application can be launched if we know where it is located: TextPad happens to be installed in **C:\Program Files (x86)\TextPad 5\textpad.exe**, but it is not necessary *type* all that: Try the following steps, which include using the **Tab** key to prompt Windows to attempt “file name completion” (do not press the **Enter** key until the step that says to do so):
  - At the command prompt type **C:\Pro** then press the **Tab** key. The command expands to **"C:\Program Files"** – not the correct folder, but it is the *first* one matching **"C:\Pro"**.

- Press **Tab** again. This time the command expands to "**C:\Program Files (x86)**" – the *second* folder matching "**C:\Pro**".
  - Then type **\Text** then press the **Tab** key.
  - Then type **\Text** and press the **Tab** key (yes, again).
  - You should now see the full path name location for TextPad, with quotes " " around it because it contains spaces. Type a **space** and then **files.txt** and finally press **Enter**.
  - TextPad should be launched, viewing/editing **files.txt**
- When you have finished with **files.txt**, you can delete it by using **del files.txt**
- Now for some navigation: View a directory listing with **dir**. Choose one of the subdirectories, say **Documents** (something else if there is no **Documents**). Type **cd Documents** to move into the **Documents** subdirectory (subfolder). **cd** stands for **change directory**. You should see that the command prompt line has changed to indicate a different working directory, and the output from **dir** will show different files/folders (or none if the folder is empty).
- Type **cd ..** to move back to the enclosing directory – verify by inspecting the command prompt and typing **dir**
- Move all the way to the top directory on the **C:** drive by typing **cd \** – verify by inspecting the command prompt. Type **dir**. Now **dir /s** will certainly give a dramatic output!
- Move directly to *any directory* on the **C:** drive by typing its location after **cd**. Using **Tab** for file name completion helps with this. Try this one: **cd C:\Windows\System32**
- Now, Windows can access file stores on various “drives”. By convention, the main hard disk drive is called **C:**, but in the University labs you probably also have **H:** connected to your personal central file store, and **V:** connected to the shared **Groups on Wide** file store. You can see which drives you have by typing the command **net use** Windows tracks a separate “working directory” on each drive. You can switch which drive you are currently working on by typing the drive letter after the command prompt, for example **H: Enter** Note, the **cd** command does not change the current drive.
- So, now switch to your **H:** drive and use **cd** to navigate to the directory where you are keeping your BlueJ Java project folders. Use **dir** to verify that you are in the right place.
- There is a quicker way to immediately open a command window at a specific folder if you can see its icon in a desktop explorer window: Hold down the **Shift** key, right-click the mouse on the folder icon and select **Open command window here** from the pop-up menu. Try it now for one of your home file store folders. Look at the working directory in the command prompt, and list the directory to verify that the command window has been opened correctly.

---

## INPUTLOOP – AGAIN!

Now you will carry out the first part of last week's lab sheet, but doing everything (almost) at the command prompt:

- Open a command window at your folder where you will keep your individual CSCU9A2 project folders, or re-use an existing command window by changing to the correct folder. If you did not create such a folder last week, then do so now either in a desktop explorer window, or using the commands below...
- Create a new directory, say **InputLoop2**, for a new project by typing the command:  
**md InputLoop2**      **md** stands for **make directory**.

- Use **cd** to move into the new directory.
- Use the **copy** command to fetch a new copy of the file **InputLoop.txt** from **Groups on Wide** and to rename it as a Java file in one step:  
**copy V:\CSCU9A2\Java\InputLoop.txt InputLoop.java**  
Note: This **copy** command does not indicate a *destination directory* (see **help copy**), so the new, renamed copy is placed in the current working directory. Remember that you can **Tab** for folder/file name completion.
- Use **dir** and **type** to make sure that the file has been copied. You should also be able to see the new folder and file in an explorer window.
- Launch **TextPad** (as above) giving **InputLoop.java** as the file to be opened. Move the window to one side for the moment.
- Compile **InputLoop.java** using the command **javac InputLoop.java** You will see many errors reported on the screen – you can scroll backwards and forwards to see them all if necessary. **javac** processes the entire file and generates as much error feedback as it can, but BlueJ cuts this down and shows us only one error at a time (that's possibly good, and possibly bad!).
- Using the TextPad window, correct the errors in **InputLoop.java**. You can **Save** from TextPad and recompile in the command window without needing to close down TextPad. At the command prompt, use the up-arrow key to select the previous **javac** command to re-execute it.
- Use **dir** to check that the **.class** file has been generated.
- Run your now correct program with the command **java InputLoop** – note that you should *not* add **.class** to the file name.
- Try some of the options that **javac** and **java** offer:
  - Type **javac** and **java** on their own at the command prompt: you will see many options listed.
  - Use **md** to make a new subdirectory called **bin**. “bin” is the traditional name for a directory containing executable files – it is short for “binary” and is a reference to machine code.
  - Direct the compiler to put the bytecode for **InputLoop** into the **bin** directory like this:  
**javac -d bin InputLoop.java**
  - Check the bytecode is there using  
**dir bin**
  - Direct the JVM to look for bytecode to execute in the **bin** directory like this:  
**java -classpath bin InputLoop**
  - Download an image from somewhere, and try the **-splash** option in **java**.

**Remember: Your code must be well formatted and clearly commented.**

**Checkpoint [Command prompt]:**

**Show a demonstrator your final code for the InputLoop program in the TextPad window, the directory listing of the bin folder, and your program running using the classpath option with java. Answer any questions they ask you.**

**For the most benefit, continue with the exercise on the next page.**

---

## INPUTWORDS

- **Working at the command prompt:**
- Create a new project folder **Words** and a subfolder **bin**, to be used as above.
- Make a copy of your **InputLoop.java** in the new folder, renaming it **InputWords.java**.
- Using **TextPad**, **javac** and **java**, design and implement the following changes to the program:
  - Modify the program so that instead of discarding inputs until an integer is found, it discards integers until a non-integer (a “word”) is found.
  - Declare an array to hold 10 **Strings** – these will be words input by the user.
  - Input 10 words from the user – the code that is in the main method, wrapped in a **for** loop and adapted very slightly, will do this.
  - Then input one further word.
  - Search for the one further word in the array of words already input. Report back to the user whether or not the word was found.
  - Make sure that your program is formatted neatly, is clearly commented, and makes appropriate use of new methods, and loops.

That’s all for this worksheet.

SBJ 22 January 2017