Using the Command Line

- Part 1 lists some common command line commands for Mac and Windows

- Part 2 is about using a text editor

- Part 3 addresses two common errors seen by students using Windows

Part 1 - Common command line commands

Mac	Windows	Description
mkdir	mk	Make a directory
cd abc	cd abc	Change to another directory
pwd	cd	Show the current directory
rm x.txt	erase x.txt	Erase file x.txt
rm -r xyz	rmdir xyz	Remove directory xyz
ls, ls -l	dir, dir/w	List the files and subdirectories in the current directory
ls ~	(no equivalent in Windows)	List the files in the user's home directory
ls /	dir C:/	List the files in the root directory
ls ..	dir ..	List the files in the parent directory
ls -l| grep -i "string"	dir | findstr -i "string"	List the files and subdirectories containing "string" in their name (Note the pipe (|) symbol is for chaining output of one command to input of the next command.)
cat	type	Display the contents of a file
cp x.txt y.txt	copy x.txt y.txt	Copy the contents of file x.txt to a new file called y.txt (Note if y.txt exists, it will be over-written.)
mv x.txt z.txt	rename x.txt z.txt	Rename file x.txt to z.txt
date	date	Display system date and time
javac	javac	Compile a Java program
java	java	Run a Java program

It is also important to know that the wildcard character (*) can be used to constrain the output of a command. For example: ls *.txt for Mac OSX, or dir *.txt for Windows will display only the files that end with '.txt'.

Part 2 - Installing and using a text editor

If you are looking for a text editor to use, read on...

Windows: You can use a tool like NotePad, or install vi or emacs. I believe you can also install Sublime (definitely can for a Mac, but I don't have a Windows machine to test it out on). Let me know if you are having trouble finding a good text editor.

Mac: When you create a file using TextEdit on a Mac, you would expect to be creating a plain text file, given the name of the application is 'TextEdit.' Unfortunately, it is actually a rich text editor (by default*), hence the .rtf default extension. (Using Microsoft Word or another rich text tool will also result in a rich text file being generated.)

Discussion: If you open an .rtf file (or .docx Word file, for example) using a true text editor, you will see something like this:

```
{\rtf1\ansi\ansicpg1252\cocoartf1504\cocoasubrtf760
{\fonttbl\f0\fmodern\fcharset0 CourierNewPS-BoldMT;}
{\colortbl;\red255\green255\blue255;\red0\green0\blue0;}
{\*\expandedcolortbl;;\cssrgb\c0\c0\c0;}
\margl1440\margr1440\vieww10800\viewh8400\viewkind0
\deftab1440
\pard\tx560\tx1120\tx1680\tx2240\tx2800\tx3360\tx3920\tx4480\tx5040\tx5600\tx
6160\tx6720\pardeftab1440\li540\fi-540\sa380\partightenfactor0

\f0\b\fs24 \cf2 \expnd0\expndtw0\kerning0
\CocoaLigature0 /* HelloWorld is the name of the Java class.\
   Every Java program needs at least one class.\
*/\
public class hw1 \{\
  public static void main(String[] args) \{\
    // Display the string\
          System.out.println(\'93Hello world!"); \
  \}\
\pard\tx560\tx1120\tx1680\tx2240\tx2800\tx3360\tx3920\tx4480\tx5040\tx5600\tx
6160\tx6720\pardeftab1440\li540\fi-540\partightenfactor0
\cf2 \}}
```

You'll recognize some of your code in there, but there is also a lot of text that you never directly typed - that other text is formatting information that's embedded in the .rtf (or .docx) file so that the TextEdit application knows how to render the text you typed - some of the formatting text is for bolding words, for selecting the font type and size, etc. Since we want plain text, with no formatting, we need to use a different text editor application.

On a Mac, I always use a tool called vi - it is not graphical, which makes it harder to learn - and you open it from the terminal window like this: vi MyNewFile.java

You could use emacs - a little friendlier than vi - but Sublime is very nice and I recommend it if you don't already know vi or emacs. In the future, we will use Eclipse, but I think it's useful to know how to create, compile, and debug programs without Eclipse.

You can download Sublime here: https://www.sublimetext.com/

After it's installed, open it as you would any other application (double click it from the Applications list on a Mac, for example). The editor should open up. Type your program in. Then select File -> Save As and enter your filename: HelloWorld.java and the directory where you want to save it.

Now that you have your file, you can continue with the steps to compile and run your program using javac and java.

Note: For Mac users only: You can change TextEdit to work as a *plain text editor* by setting configuration options as described below, however, Sublime is preferred because it provides language-sensitive highlighting which will be helpful as you develop your code.

Follow these steps if you prefer TextEdit to Sublime:

1. Open TextEdit

2. Select TextEdit > Preferences

  Format > Plain Text

  Font > Plain Text Font: A fixed point font like Courier New is a good idea

  Options > I de-selected everything (see attached screenshot)

3. Save as .java file

Part 3 - For Windows machines, we typically need to add the path to the Java compiler to the path environment variable

Two types of errors you may see:

Error 1: `'file not found'`

This error is likely because you have issued the **java** command from a directory where the Java compiler is not installed.

If you created your Java program, e.g. **HW1.java,** in directory **c:\NYU_Java\Homework_1** use the **cd** command (change directory command) to **cd** to the directory where your file is stored. For this example, type this:

cd c:\NYU_Java\Homework_1

_____

Error 2: `'javac'` is not recognized as an internal or external command, operable program or batch file

This error is likely because you have issued the **javac** command from a directory where the Java compiler is not installed.

To correct the problem, you need to set the path environment variable so that the compiler can be found when invoked.

You can set the path using the following steps:

1) Open System properties (run `sysdm.cpl`)

2) Go to ***Advanced*** tab

3) Click on ***Environment variables***

4) Under ***System Variables*** scroll down to find `PATH`. Edit the entry and add the path to the compiler.

   For example, if the JDK path is C:\Program Files(x86)\Java\jdk1.8.0_121\bin, enter this path.

   (Note: the bin subdirectory is where java and javac reside.)

5) Click OK and close the windows.

6) Open a new command prompt and run the following

`set path`

   Look carefully at the output of the command; you should be able to see the JDK path you just added.

   Note that if you run the command from a window which was opened before you modified the PATH variable, it will print the old path.

7) Now try compiling and running your program:

`javac HelloWorld.java`

`java HelloWorld`