Course Preliminaries

Syllabus:
https://pilot.wright.edu/d2l/le/content/264966/viewContent/900316/View

Programming Assignments:

1. All programming must be submitted via the dropbox.
2. Programs will be compiled and tested using GNU g++ compiler on unixapps1.wright.edu
3. Students must provide all source files and a makefile that can be used to build each assignment.
4. Any software solution submitted without a working makefile or that requires editing to make the source code compile will receive significant reduction in credit.

Cooperation vs Plagiarism:

1. Students are encouraged to form study groups, share ideas and in general help each other.
2. You may NOT use other students (past or present) software in your solutions.
3. You may use publicly available algorithms to help you solve your assignments.
4. If you use an algorithm not taken from the textbook or the lecture, you must include a complete citation in your code indicating where you obtained the algorithm.
5. If you use some else's algorithm, you must be able to explain how it works to the instructor's satisfaction.
6. If you are not sure if something represents plagiarism, ask the instructor -- don't assume it is okay!

Sunday, August 24, 2014 8:34 PM

Accessing unixapps1.wright.edu

You should already have an account that uses your W-number and campus password

You will need some tools depending on where you plan to work.

- If you plan to work on campus using a hardwired system (e.g. open lab on the 1st floor of Russ), you should not need to install any additional software. You can login and get started.

- If you plan to use a wireless connection on a personal laptop, when you are on campus you will need to login via the secure network.

- If you plan to work from home you will need to VPN: http://www.wright.edu/information-technology/security/virtual-private-networks-software-overview

- If you plan to work from your own laptop or desktop, you will need to install an FTP application and a terminal application. What you use depends on your operating system (Windows/Mac). You may already have a suitable application on your machine, but if you do not have something available, you can download a free product from CaTS (see specialty software): http://www.wright.edu/information-technology/services/free-software-connectwright

  (I use SSH on a Windows machine as well as Notepad++.) There are also other free products available on the web.

  Now let's login to unixapps1

Basic Unix Command

There are many source where you can learn about Unix commands.

http://mally.stanford.edu/~sr/computing/basic-unix.html

http://www.math.utah.edu/lab/unix/unix-commands.html

http://www.math.harvard.edu/computing/unix/unixcommands.html

https://kb.iu.edu/d/afsk

When you are logged onto the Unix machine, and you know the name of the command; you can always get information about a command using the manual page: man command-name

By default, when you login you will be running the tcsh shell. Although you don't need to do much with the shell, you can use the shell to customize your work environment and automate certain tasks. For more information, you can tcsh site.

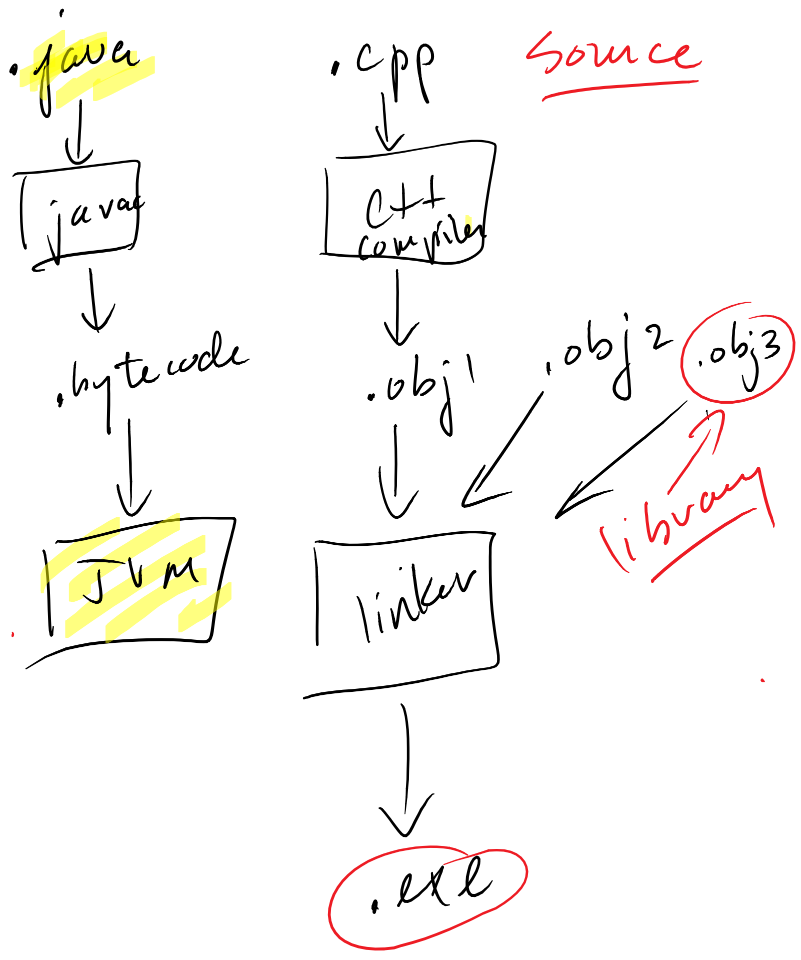
http://www.tcsh.org/Home

Sunday, August 24, 2014 9:46 PM

make utility

SAMPLE Makefile

```
CFLAGS     = -g
OBJFILES   = Stack.o Queue.o hw1.o

#####################################################
# Default rules for compiling C++ programs     #
#####################################################
.cpp .o:
        g++ $(CFLAGS) -c $*.cpp

hw1:       $(OBJFILES)
        g++ -o hw1 $(CFLAGS) $(OBJFILES)

hw1.o:     hw1.cpp Queue.h Stack.h

Stack.o:   Stack.cpp Stack.h

Queue.o:   Queue.cpp Queue.h

clean:
        rm -f *.o
```



Notes

Make manual:

http://www.gnu.org/software/make/manual/

Notation:

Target: Dependency

Define a variable and its associated value:

variable = value

Use a variable - value in a rule:

$(variable)

Built-in targets:

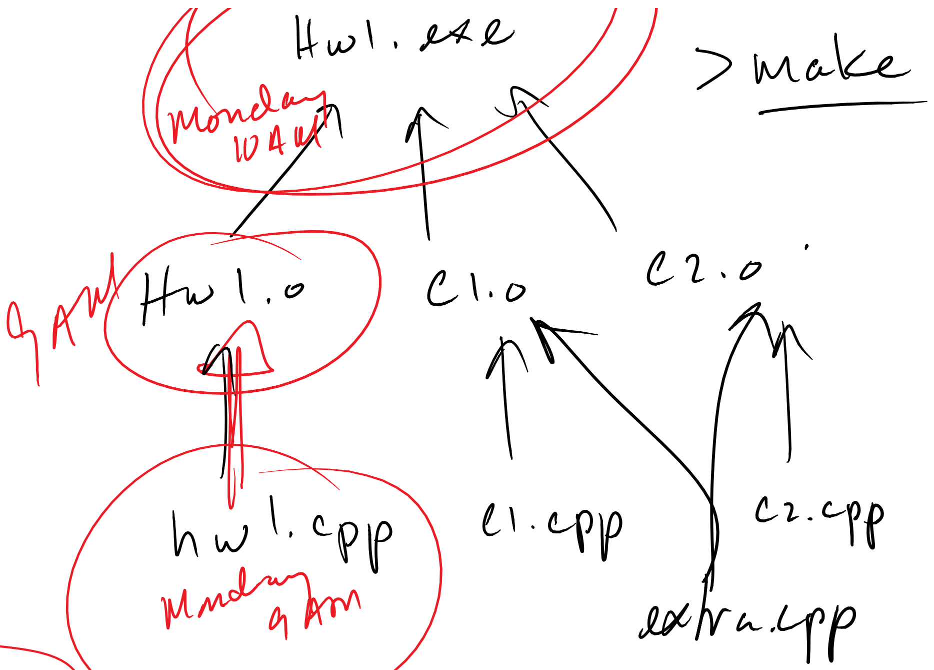
http://www.gnu.org/software/make/manual/make.html#Special-Targets

Automatic variables:

http://www.gnu.org/software/make/manual/make.html#Automatic-Variables

Hw1.exe ✓

> make



```
-o hw1
> g++ hw1.cpp c1.cpp c2.cpp extra.cpp
      hw1.o   c1.o   c2.o   ~~extra.o~~
      ~~a.out~~   hw1

> ~~a.out~~ hw1

> g++ -c hw1.cpp
           ↓
         hw1.o

> g++ -o hw1 hw1.o c1.o c2.o extra.o

> make
```

```
// C++ Program

#include <iostream>

using namespace std;

int main( int argc, char* argv[] ) {

   int x = 100;

   cout << "x= " << x << endl;

   return 0;
}
```

Useful site will C++ reference and tutorials:

http://www.cplusplus.com/