Oregon State University
School of Electrical Engineering and Computer Science

CS 261 – Recitation 1
Compiling C on UNIX

Winter 2015

Outline

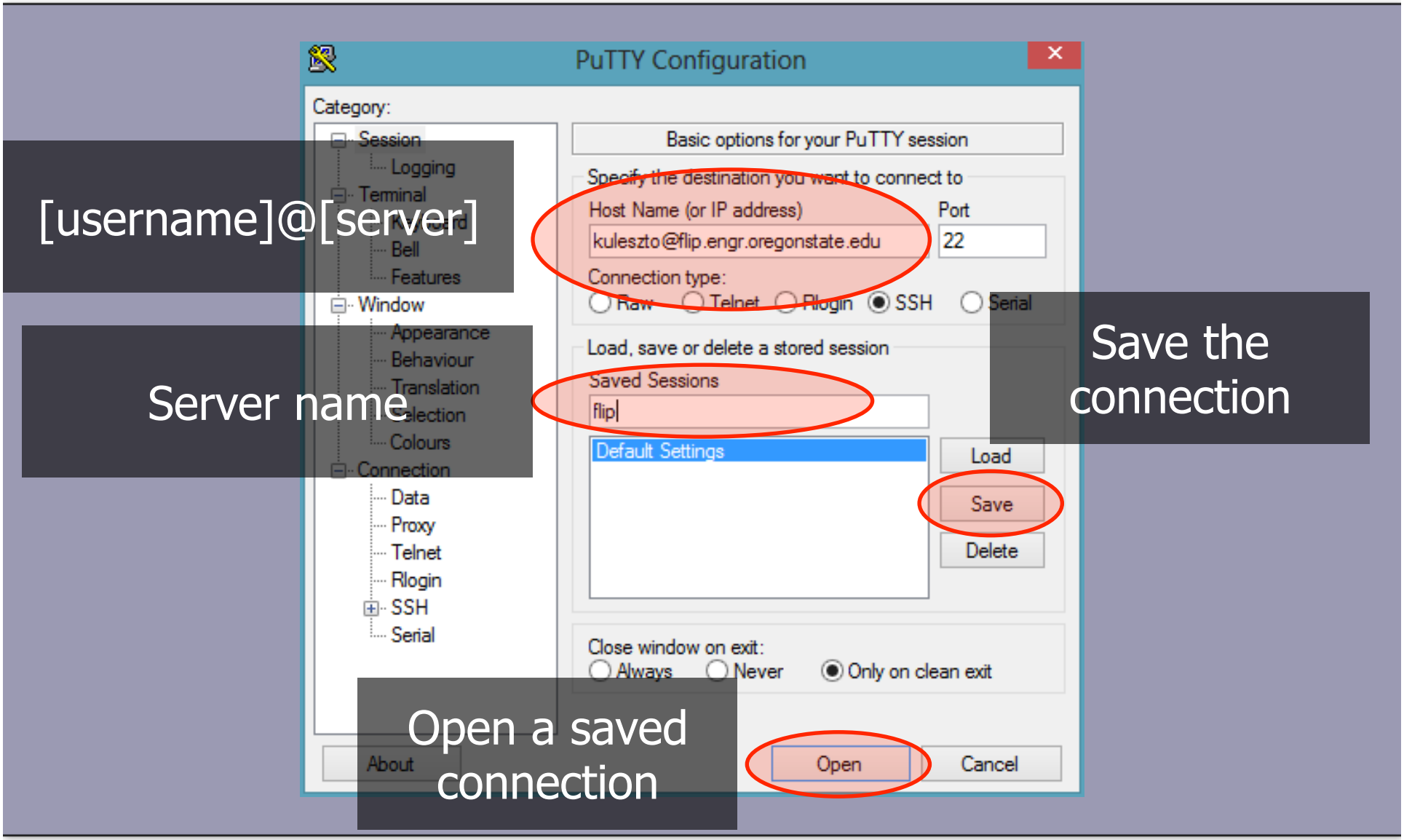
- **Secure Shell**
- Basic UNIX commands
- The GNU Compiler Collection (gcc)
- Setting up your IDE

Downloads: http://dropline.net/cs261

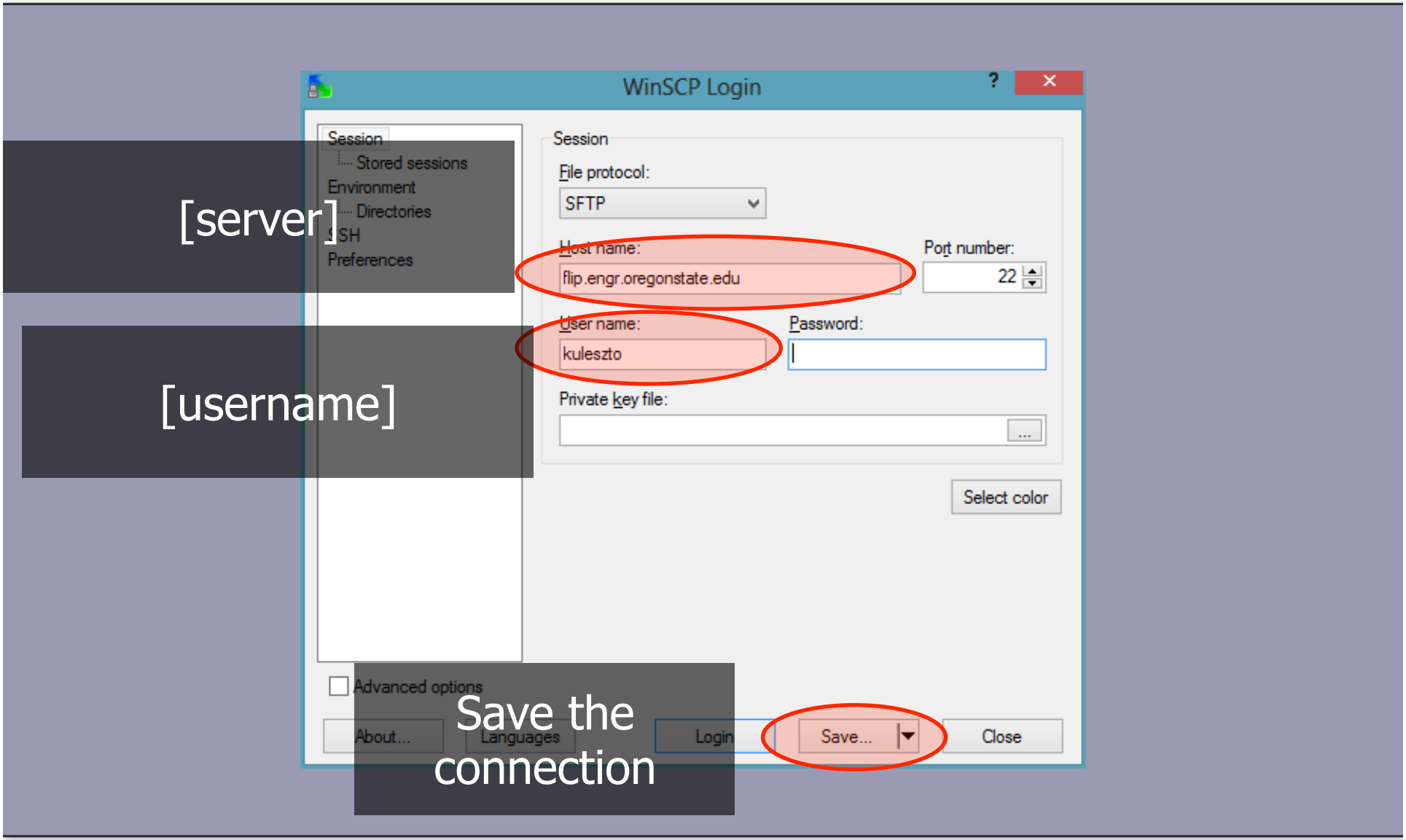
Secure Shell Tools

- Mac OS X / Linux
  - Open a terminal
  - Type "ssh [username]@[server]"
    - *ssh kuleszto@flip.engr.oregonstate.edu*
  - scp [file] [username]@[server]
    - Transmit, Cyberduck, Filezilla, etc. (if you want a GUI)
- Windows
  - Putty (http://engineering.oregonstate.edu/computing/fileaccess/putty_ssh/)
  - WinSCP (http://winscp.net/)

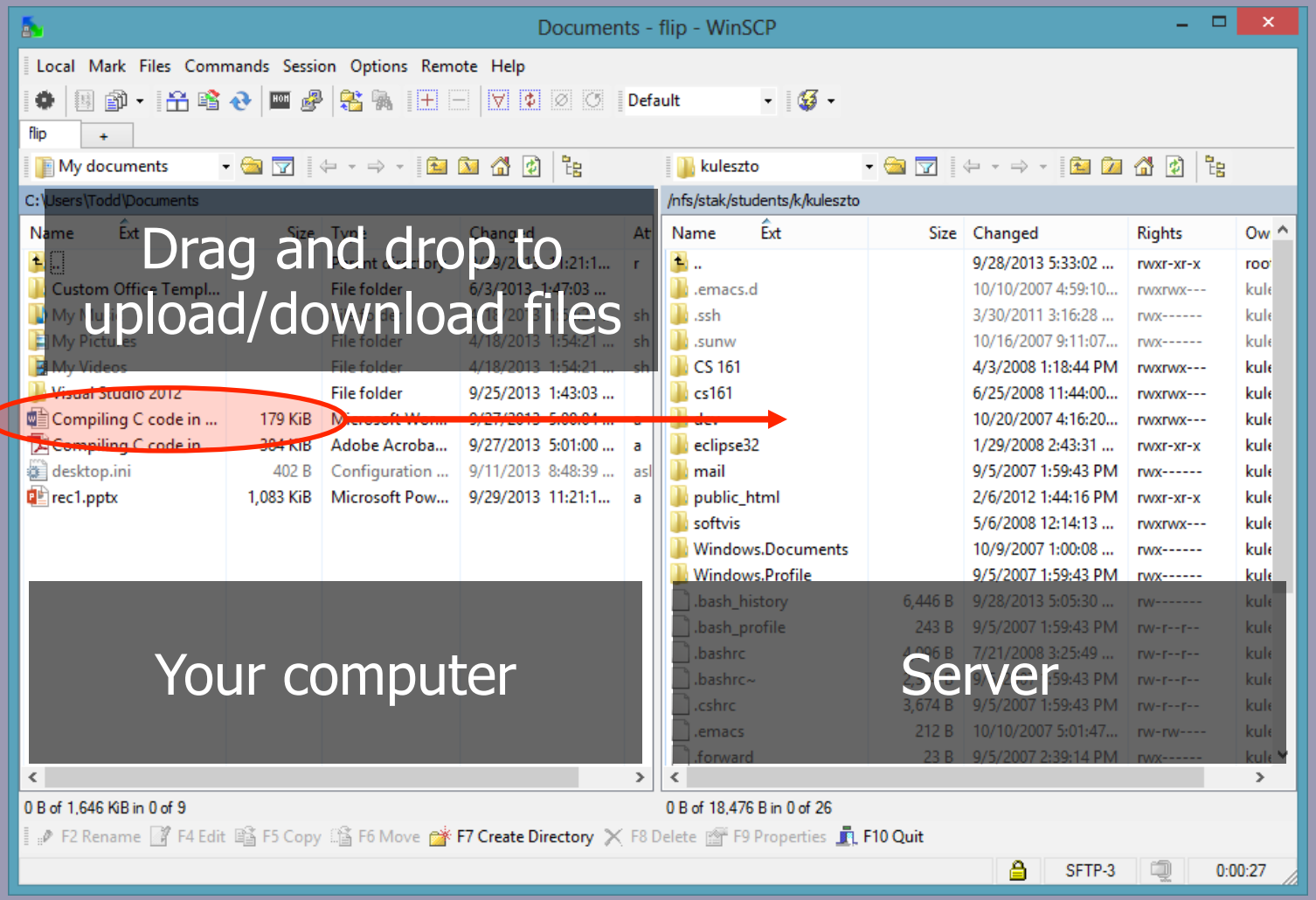
Putty Configuration



WinSCP Configuration



WinSCP Usage



UNIX Servers @ OSU

- ENGR
  - flip.engr.oregonstate.edu (RHEL 6.5)
  - flop.engr.oregonstate.edu (RHEL 6.5)
- ONID
  - shell.onid.oregonstate.edu (Debian 6.0)

Outline

- Secure Shell
- **Basic UNIX commands**
- The GNU Compiler Collection (gcc)
- Setting up your IDE

Basic UNIX Commands

Command	Description
`ls`	Lists files and folders (use "ls -l" for a "long" listing)
`cd [dir name]`	Change directory ("cd .." will go up a directory)
`pwd`	Print name of current directory
`mkdir [dir name]`	Make a directory
`rmdir [dir name]`	Remove a directory
`cp [file] [new file]`	Copy a file (use "cp –r" to copy a directory)
`mv [file] [new file]`	Move (or rename) a file or directory
`rm [file]`	Remove a file
`cat [file]`	Show file contents
`exit`	Close connection to server
`ctrl-c`	Kill the current process

Basic UNIX Commands

- For more info on a command, use the manual page:

  $ man ls

- Tutorial: http://www.ee.surrey.ac.uk/Teaching/Unix/

Outline

- Secure Shell
- Basic UNIX commands
- **The GNU Compiler Collection (gcc)**
- Setting up your IDE

GCC

- GCC is the standard UNIX/Linux/Mac OS X compiler for about a dozen languages (e.g., C, C++, Objective C, Fortran, Java)
- Compiling with GCC:

  gcc <list of options> sourcefile.c

  e.g.: *gcc -Wall -std=c99 -o test test.c*
- Compiling multiple files:

  gcc <list of options> <list of source files>

  e.g.: *gcc –Wall -std=c99 -o test test1.c test2.c test3.c*

Makefile Example

A makefile is like a script for the compiler. We'll provide makefiles for many of your assignments so you'll be able to compile by typing 'make'.

Contents of a makefile:

```
default:main


main: main.c
   gcc -Wall -std=c99 main.c -o main
clean:
   rm main main.o
```

Executed when you type "make"

Executed when you type "make clean"

Make Tutorial

http://mrbook.org/tutorials/make/

Practice

Follow the instructions under
"Practice working from the command line"
from
http://classes.engr.oregonstate.edu/eecs/winter2015/cs261-001/lab1.php

Outline

- Secure Shell
- Basic UNIX commands
- The GNU Compiler Collection (gcc)
- **Setting up your IDE**

C Programming IDEs

- Mac only: **Xcode**
- Windows only: **Visual Studio**

Always Test on UNIX!

You can use any IDE to develop and test your C application before submitting. However, UNIX is the environment in which the program will be graded.

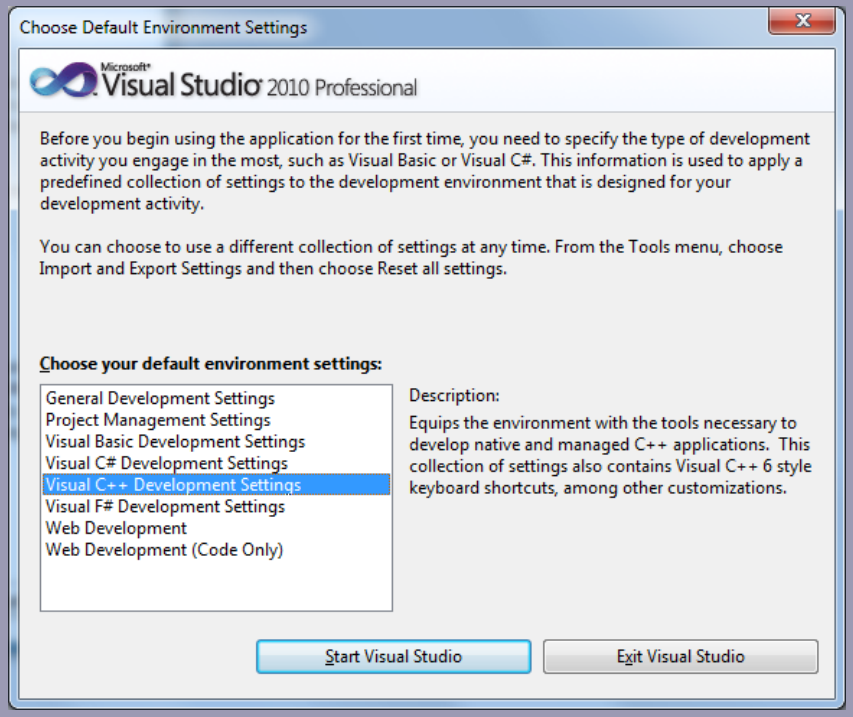
Make sure your program will **compile and run without errors or warnings** using GCC on '***flip.engr.oregonstate.edu***'.

Microsoft Visual Studio 2013

- Download it from https://secure.engr.oregonstate.edu:8000/teach.php?type=want_auth
  - Click on "Microsoft Dreamspark Login"
  - Visual Studio is under "Development Tools"

Setting up Visual Studio

- First time you start MSVS
  - The first time you start visual studio it will ask what environment settings to use.
  - Select "Visual C++"
  - On EECS lab machines you may wait a loooong time.



Setting up your IDE

See our handouts on Visual Studio and Xcode

That's all for today!