## Common OS Controls:

Move the cursor per word -> CTRL + LEFT/RIGHT ARROW KEY

Select the whole word -> CTRL + SHIFT + LEFT/RIGHT ARROW KEY

Remove the whole word -> CTRL + BACKSPACE

Reach the End of Line -> END KEY

Reach the Beginning of Line -> HOME KEY

Select whole line -> SHIFT + HOME (If cursor is at the end of line.)

Select whole line -> SHIFT + END (If cursor is at the beginning of line.)

Select entire document characters -> CTRL + A

Open My Computer/THIS PC -> WINDOWS + E

Open Run -> WINDOWS + R

Minimize all the windows programs -> WINDOWS + D

Fast scrolling browser’s web pages or documents: Make use of PgUp/PgDn, HOME/END keys.

CTRL + T - New Tab

WINDOWS + L to lock computer

F5 to refresh

Control + U to lowercase all highlighted letters but add SHIFT U if wanting to capitalize it

Alt + MOUSE DRAG to highlight specific area to delete

//==============================

## Common CLI Commands for Window Command, Git Bash, & Powershell:

pwd - print working directory

cd - change directory

dir - list files in directory

copy con - create empty file

mkdir - create empty folder

cp - Copy folder [Syntax - cp existingFolder NewFolderName]

vi fileName.cs //Alters internal component like a text editor

ls -l //Lists all files in directory

- Strictly for Git Bash in Commit Mode

i //enter inline insert mode

:x! //: goes into command while x! means quit/save

javac \*.java //Compile all java source files

//==============================

## Github/Git Bash Cheatsheet:

#### Git Global Credentials . . .

Set global Credentials

git config --global user.name "John Doe"

git config --global user.email doe123@mail.chapman.edu

#### Git Clone . . .

git clone [LINK] [Directory or Folder Name]

NOTE:

<https://ogfeng-josh@github.com/ogFeng-Josh/ITN.git> //for private repository with @

If folder is not empty...

git init

git remote add origin PATH/TO/REPO

git fetch

git checkout -t origin/master

If cloning into empty folder

git clone git@github.com:whatever .

#### Git Commit Changes . . .

git add [newFileinDirectory] //Adds new file in staging area (git add . = adds all file)

git commit -m "Message" //Commit local push

git commit -a //Commit all changes to included files to staging area (Doesn't include new files so use git add...)

git push -u origin <branch> //Pushes new branch & changes to new branch

git push //Pushes changes

#### Git Branch Control . . .

git checkout [BRANCH NAME] //Switches to branch

git checkout -b mybranch to create and switch to the branch "mybranch" on your local repo

git branch -a //list all branches and current (Highlighted in green)

git branch -d <local branch name> //Deletes local branch

Git pull origin [Branch name] //Updates branch from online repository

#### File Control . . .

py -m file.py //Runs python file | Replace py with language type

./checkstyle.sh \*.java OR checkstyle.bat \*.java //Checkstyle for file

#### Git Commit Documentation:

<https://git-scm.com/docs/git-commit>

//==============================

## Docker Toolbox Start Up:

Start Linux Container...

docker start compsci

Attach Linux Container...

docker attach compsci

Once we see root command line, we check directory by...

ls = List files in directory

cd /home/directorypath = Set directory of project files

Name project files...

touch foo.txt = create txt file and name it foo

g++ filename.cpp = compile FIRST

OR

For C++

Remember this format

g++ -o [Insert name to run] [file type]

Then ./[name] to run

Ex)

G++ -o MyProj main.cpp source1.cpp head1.h

Check with ls

./MyProj to run

For Java

Javac \*.java //Compiles Java code OR to compile a specific file

Javac MtClient.java < (file name)

Then to run

Java (name of program)

//To create a docker image

Docker image build -t (name) .//Creates temporary container

Then

Docker container run --rm -t --name red(client name) (image name)

Winpty docker container run --rm -t -name blue(client name) (image name)

//For Docker Container containers, List docker networks & show details of bridge networks

Docker network ls

Docker network inspect bridge (BRIDGE is a network name, replace if different name given)

./file name with OUT = execute program

./a out //Runs program OR ./NAMEOFPROGRAM (Anything else required for input)

g++ [file name] -o mytest.out

Finish & Exit Linux Container...

exit

Note: for Gitbash when needed for TTY error (When using windows only)

Type “winpty” before command

To use make fill type in command “make all” when you have a make file set up

//==============================

## VS Studio Code Debug Syntax:

Controls to Remember:

CTRL + X = Delete Line

CTRL + A = Select all words available

CTRL + (Left/Right Arrows) = Navigate by whole words without highlighting like CTRL + Shift + ArrowKeys

CTRL + Enter = Create new line before current line

Highlighted Section + TAB = Indent Lines

Highlighted Section + Shift + TAB = Un-Indent Lines

CTRL + F = FInd in specific CS file

CTRL + Shift + F = Find in whole project

Alt + Shift + ArrowKeys to highlight section of words/lines

CTRL + K + C/U = Comment/Uncomment out code

CTRL + K + K = Bookmark Line/Section of code

CTRL + T = Search every CS or available file for keyword (A global CTRL F, USE THIS!)

Highlight + ALT + Enter = Remove unused directive/redundant code (DONT USE UNLESS YOU KNOW HOW IT WORKS)

Debugging Controls to Remember:

Delete/Disable all breakpoints Option in Debug Dropdown Menu

CTRL + Shift + B = Test application for errors by building

CTRL + F5 = Run Application without Debug Mode (Normal)

F5 = Run Application in Debug Mode

F5 (WHEN ALREADY RUNNING) = Continue Running Application till next Breakpoint set up or end of program.

Shift + F5 = Stop Debugging Application

CTRL + Shift + F5 = Restart Debug Application

F9 = Create Breakpoints (Red Dot on left side of # column)

F10 = Execute code line by line. When at breakpoints, use this to execute next line and drag yellow arrow back if you want to backtrack

F10 = Step over code (If you were to call a method/function and not want to enter it)

F11 = Step Into code (As explained above but you want to enter said method/function)

Shift + F11 = Step out (Stepping out of current method/function you had stepped into)

//==============================

HTML SSH to www1 Server:

Ssh to www1.chapman.edu server and use student portal password to login

Cd to whatever directory (Usually public\_html)

Mkdir “name” //Makes new directory

Vi “File name” to create file

Then ESC & :x to save and quit

//==============================

Using C++ with VSCode:

Link: <https://nuwen.net/mingw.html>

Install minGW and check with g++ --version

Then open VSCode & install Code Runner by Jun Han

Finally, install C/C++ IntelliSense by Microsoft

Now starting it by creating & saving file as C++ first.

Then compile

Remember . . .

Run code by Ctrl + Alt + N OR Press F1 + select/type Run Code

Or you can right click text editor & Run code

Or click Run Code in editor title menu or context menu

To Stop code, press

Ctrl + Alt + M

F1 + Select/type Stop Code Run

Click Stop Code Run in context menu of output channel

//==============================

Using C# with VSCode:

Link: <https://dotnet.microsoft.com/download>

Install .NET package and confirm installer

Check version by dotnet --version to confirm

Now install C# for VSCode by Microsoft

When starting, create project by going to folder of project, create new project directory (Like Visual Studio Code sln file)

Type in Git or CMD “dotnet new console”

And wait for files to download

Run by typing in console

dotnet run

Set

cd $dir && dotnet run $fileName

If you want to click and run using code runner

C++:

Print each letter at a time

//Print Screen delays each character printout to simulate a terminal screen

// void PrintScreen(const std::string& str/\*, int delay\_time\*/)

// {

// for (size\_t i = 0; i != str.size(); ++i)

// {

// std::cout << str[i];

// Sleep(1);

// }

// //std::cout << std::endl;

// }