[Note: If you are using the Ubuntu VM provided by the instructor, these instructions have already been executed for you!]

Ubuntu VM installation and setup

1) Download and install the VirtualBox platform package for your operating system, as well as VirtualBox Extension Pack (current version is 5.2.4): https://www.virtualbox.org/wiki/Downloads

2) Download the Ubuntu virtual disk (VDI) for VirtualBox (current version is 17.10):
https://www.osboxes.org/ubuntu/

3) In VirtualBox, create a new Ubuntu VM. At the "Hard disk" step, choose "Use an existing virtual hard disk file" and navigate to the Ubuntu VDI that you downloaded in the previous step.

4) Start your new Ubuntu VM (the default username and password are both "osboxes.org").

5) In a Terminal window, type the following commands to bring your installation up-to-date, and to install gcc and associated build tools, header files, and the Revision Control System (RCS) <https://www.cs.colostate.edu/helpdocs/RCS.html>:

```
sudo apt update

sudo apt dist-upgrade

sudo apt install build-essential

sudo apt install linux-headers-4.13.0-21-generic

sudo apt install rcs
```

6) Install pip <https://pip.pypa.io/en/stable/>, and gdbgui <https://gdbgui.com/>:

```
sudo apt install python-pip

sudo pip install gdbgui --upgrade
```

[Optional:] Code editor installation

- To install Emacs <https://www.gnu.org/software/emacs/>:

```
sudo apt install emacs25
```

- To install Vim <https://www.linux.com/learn/vim-101-beginners-guide-vim>:

```
sudo apt install vim
```

- To install Visual Studio Code: Download the appropriate file from https://code.visualstudio.com/#alt-downloads

After the .deb file is downloaded, in Files navigate to Downloads, right-click on the code[...].deb file, and choose "Open with Software Install".

Compiler/interpreter installation

- Ada (using the "gnatmake" wrapper for gnat, the gcc Ada compiler front-end) <https://docs.adacore.com/gnat_ugn-docs/html/gnat_ugn/gnat_ugn/building_executable_programs_with_gnat.html>:

```
sudo apt install gnat-6
```

- C# <http://www.mono-project.com/>:

```
sudo apt install mono-devel
```

- Go <https://golang.org/>:

```
snap install --classic go
```

- Haskell <https://www.haskell.org/>:

```
sudo apt-get install -y software-properties-common
sudo add-apt-repository -y ppa:hvr/ghc
sudo apt-get update
sudo apt-get install -y cabal-install
sudo apt-get install -y ghc
sudo apt-get install -y haskell-stack
stack upgrade

cat >> ~/.bashrc <<EOF
export
PATH="\$HOME/.cabal/bin:/opt/cabal/1.22/bin:/opt/ghc/7.10.3/bin:\$PATH
"
EOF
export PATH=~/.cabal/bin:/opt/cabal/1.22/bin:/opt/ghc/7.10.3/bin:$PATH
```

- Java (using jGRASP) <http://www.jgrasp.org/>:

```
sudo apt install default-jdk
```

```
sudo apt install lsb-core

Download jGRASP Zip file from:
http://spider.eng.auburn.edu/user-cgi/grasp/grasp.pl?;dl=download_jgra
sp.html

Extract the jGRASP Zip file to /home/osboxes .

export PATH=/home/osboxes/jgrasp/bin:"${PATH}"
export PATH=/home/osboxes/jgrasp/jbin:"${PATH}"
```

- Prolog <http://www.swi-prolog.org/>:

```
sudo apt install swi-prolog
```

- Python <https://www.python.org/>:

```
(python2 and python3 are pre-installed on the OSBoxes Ubuntu image)
```

- OCaml <https://ocaml.org/>:

```
sudo apt install ocaml
```

- Ruby <https://www.ruby-lang.org/en/>:

```
sudo apt install -y software-properties-common
sudo add-apt-repository -y ppa:brightbox/ruby-ng
sudo apt update
sudo apt install ruby2.4
```

- Rust <https://www.rust-lang.org/en-US/>:

```
sudo apt install curl
curl https://sh.rustup.rs -sSf | sh
```

- Scheme <http://racket-lang.org/>:

```
sudo add-apt-repository ppa:plt/racket
sudo apt update
sudo apt install racket
```

- Swift <https://developer.apple.com/swift/>:

```
Download the Swift 4.0.3 distribution:
https://swift.org/builds/swift-4.0.3-release/ubuntu1610/swift-4.0.3-RE
LEASE/swift-4.0.3-RELEASE-ubuntu16.10.tar.gz

and the signature file:
https://swift.org/builds/swift-4.0.3-release/ubuntu1610/swift-4.0.3-RE
LEASE/swift-4.0.3-RELEASE-ubuntu16.10.tar.gz.sig
```

```
sudo apt install clang libicu-dev --fix-missing
wget -q -O - https://swift.org/keys/all-keys.asc | \
  gpg --import -
gpg --keyserver hkp://pool.sks-keyservers.net --refresh-keys Swift
gpg --verify swift-4.0.3-RELEASE-ubuntu16.10.tar.gz.sig
mkdir /home/osboxes/Swift
cd /home/osboxes/Swift
tar xvf ~/Downloads/swift-4.0.3-RELEASE-ubuntu16.10.tar.gz
export PATH=/home/osboxes/Swift/usr/bin:"${PATH}"
```

[Optional:] Additional language environment installations

- Common Lisp (SBCL) <http://lisp-lang.org/learn/getting-started/>:

Install the SBCL environment, the Quicklisp package manager, and SLIME (a Common Lisp IDE built on Emacs) as follows:

```
sudo apt install sbcl
curl -o /tmp/ql.lisp http://beta.quicklisp.org/quicklisp.lisp
sbcl --no-sysinit --no-userinit --load /tmp/ql.lisp \
     --eval '(quicklisp-quickstart:install :path ".quicklisp")' \
     --quit
```

Add the following to your ~/.emacs file:

```
(load (expand-file-name "~/.quicklisp/slime-helper.el"))
(setq inferior-lisp-program "/usr/bin/sbcl")
```

To run Common Lisp from the command line, simply type “sbcl” in a Terminal window.

To run SLIME, start Emacs and type “M-x slime” (that is, the Alt key along with the x key), then type “slime” in the little buffer at the bottom. Press enter, and a REPL will start.

- Erlang <https://www.erlang.org/>:

```
wget https://packages.erlang-solutions.com/erlang-solutions_1.0_all.deb
sudo dpkg -i erlang-solutions_1.0_all.deb
sudo apt update
sudo apt install esl-erlang
```

- F# (using mono) <http://www.mono-project.com/>:

Install mono (if not already installed for C#) following the directions at http://www.mono-project.com/download/

Then follow the additional instructions to install F# at

https://medium.com/@edgarsanchezg/four-easy-steps-for-installing-f-in-ubuntu-linux-88122323f12b

- Fortran <https://gcc.gnu.org/fortran/>:

```
sudo apt install gfortran
```

- Icon <https://www2.cs.arizona.edu/icon/>:

Follow the directions in <https://www2.cs.arizona.edu/icon/current/>.

- Modula-3 (using cm3) <http://modula3.org/>:

Download the AMD64_LINUX standard release here:
http://www.opencm3.net/releng/cm3-bin-core-AMD64_LINUX-5.8.6-REL.tgz

Extract the directory, then run the "cminstall" script using sudo. Follow the directions given by the installation script to add the resulting bin directory to your PATH.

- Pascal (using Free Pascal) <https://www.freepascal.org/>:

```
sudo apt install fp-compiler
sudo apt install fp-ide
sudo apt install fp-docs
```

Run the Free Pascal compiler by typing "fpc file.pas". Run the terminal-based IDE by running "fp" in Terminal.

- Smalltalk (using Squeak) <http://squeak.org/>:

Follow the instructions for your OS in <http://squeak.org/downloads/>.

Besides the standard Squeak image, you can also try out other "project" images such as "etoys" and the original MIT Scratch implementation; see <http://squeak.org/projects/>.