Commands:

Basic navigation:

ssh - secure shell, log into a machine
clear - clears the terminal of previous commands
bash - enters into the bash shell (default on ND machines is -tch)
echo - echo typed words, or a variable (which has $ before it)
declare - defines a variable, i.e. declare myvar=hello. No spaces permitted without escape or quotes.
pwd - displays the absolute path to the current directory from root (/)
ls - listing or "let's see". lists the files in the current folder.
man COMMAND - displays the manual pages for the command
cd - change current directory
mkdir - make a new directory
less FILE - opens a readable file, does not allow you to edit it.
nano - opens a file (or creates a new one) in a word pad style editor
mv FILE DESTINATION - moves a file from one point to another
mv FILE NEWNAME - renames a file
cp FILE DESTINATION - copies a file from one point to another
rm FILE - removes a file (-r flag needed to remove a directory)
scp FILE COMPUTER:DESTINATION - moves a file from home computer to another computer, in the file designated after the :
scp COMPUTER:FILE DESTINATION - moves a file from another computer to the current computer.
top - list the users on the machine and the resources in use
head - list out the top ten lines of a file to the terminal
tail - list the last ten lines of a file to the terminal
info COMMAND - more extensive info than on the man pages
date - outputs date and time
which COMMAND - tells you where a program exists if it is within you PATH variable
ln -s DESTINATION LINKNAME - creates a symbolic link from the destination (ie:dropbox in course directory) to a ./linkname/ (ie:linktodropbox). Deleting a softlink does not effect the actual directory (unless you use rm -r).
gzip FILE - compress a file
gzip -d FILE - decompress a file, file must end in .gz
tar cf FILE - tar files together
tar xfz FILE - untar files: e(x)tract (f)rom g(z)ip

File System/System Information:

free -mg - gives you the amount of RAM on the current machine in GB (-g)
du -h - gives you disk usage information (in human readable format)
df -h - gives you info about amount of disk free (in human readable format)
top - shows all processes running and who owns them, their CPU usage, Memory usage, and other things (q to quit)

Obtaining Data and Software:

wget http://... - downloads file at http link (can also use with ftp link)
gzip -d file.gz - decompresses file.gz (can be a tar.gz file)
bunzip2 file.bz2 - decompresses file.bz2 (can be tarred)
unzip file.zip - decompresses file.zip (can be tarred)

tar -xf file.tar - unpacks file.tar
tar -zxf file.tar.gz - decompresses and unpacks a tar.gz file
tar -zcvf file.tar.gz targetdir - makes file.tar.gz out of targetdir (z = gzip, c = create)

Installing Software:

./configure --prefix=/path/to/install - configures your source to install. Leave off prefix if you have root and want to install to /usr/bin
make - creates binaries
make install - installs to the path specified in prefix
arch - check to see if we're running a 32- or 64-bit machine

Special Characters:
/ - separates directory names in a file path (ex: MyDocuments/MyMovies/movie.mov). Also refers to "root" directory (/)
\ - "escape character", causes the single character following it to be read literally (ex: echo \\ echoes a single \) '
- used in pairs to take a phrase and use it and its contents literally (ex: echo 'Yay bioinformatics')
" - used in pairs to take a phrase and use it and its contents literally except for ! and $, to allow variable subsitution (ex: echo "My path is $PATH")
$ - signals a variable. When inside a set of "" CANNOT be escaped and always signifies a variable. (i.e. echo "$HOME is my home")

* - used to match any number of characters. i.e. test*.txt, test*, *
? - used to match any single character. i.e. test1?.txt, test?.txt
; - used to signal the end of a command on a single line.
- designates a comment in bash. Anything following will not be read by computer.
#! - hash bang or shebang, tells the computer you are about to give it a program to use to interpret the code you are providing. i.e. #!/bin/bash
: - used to separate directory locations in PATH variable.
> - used to capture output to screen and put it into a file.

Hints:

Tab complete - will complete names to the next point of ambiguity
Hit the up key to go back through your previous commands
You can chain together multiple directories for commands. Ex. cd ../../.. - goes up 3 directories
Ctrl-c kills a running program
Add directories to your PATH with "declare PATH=$PATH:
.bashrc is your set up for your terminal in bash. You can add any code in this file that you wish to be executed upon entering bash (such as adding things to your path, aliases, variable declarations, etc).