

The terminal is an extremely powerful way to interact with your computer. Below are some very common commands you should know and will use throughout the semester. Items encased in "less than",  $<$ , and "greater than",  $>$ , symbols denote user-defined (you make them up) inputs, i.e.  $<filename>$ .

Command	Common Pronunciation	Description
ls	"el-es"	list all files and directories (folders) in the current directory.
mkdir $<dir\_name>$	"make dur"	create a new directory in the current directory
cd $<path>$	"see-dee"	Change directory to the one supplied in $<path>$
g++ $<arguments>$	"see-plus-plus"	Compile a C++ source file according to the arguments provided greater detail for this command included below.
cat $<file\_name>$	"cat"	displays the contents of the given file to the terminal
cp $<filename>$ $<destination>$	"see-pee"	Copy file $<filename>$ to directory $<destination>$ . Name of file copy can be included in the destination directory provided.
rm $<path>$	"ar-em"	Remove item given by path. I suggest running "man" on this command to see all of the ways this can be used.
touch $<file\_name>$	"tuhch"	Create an empty file named $<file\_name>$ . Don't forget the file extension!
./ $<compiled\_prog>$	—	run the compiled program. Program must be in the current directory

# Breaking Down Our Compile Command

`g++ -Wall -Wextra -pedantic <source file> -o <output_name>`

Diagram illustrating the components of the compile command:

- `g++`: Name of our compiler
- `-Wall`: display all warnings
- `-Wextra`: display extra warning info
- `-pedantic`: ensure source code is compliant with C++ standard
- `<source file>`: Name of the file to compile (don't forget the extension)
- `-o`: Set output file name to next argument supplied
- `<output_name>`: output file name, no extension needed!

Space delimited