VI HANDOUT: FREQUENTLY ASKED QUESTIONS
AND USEFUL COMMANDS

"Help! I can’t type anything! It’s just beeping at me."

Vi is a dual-mode editor, which means that it has a command mode and an insert mode.

**Command mode:**
When you start vi, you start in command mode. You cannot insert text while in command mode. To change to insert mode, you must use an insert command. To get back into command mode at any time, press the escape key (ESC on most keyboards.) If you press ESC to get into command mode, but are already in command mode, you will hear a beep. This just means that you were already in command mode. So when in doubt, press ESC.

**insert commands:**
i     inserts text before the cursor
a     appends text after the cursor
0     opens a line for writing immediately below the current line
I     inserts text at the beginning of the line
A     appends text at the end of the line
O     opens a line for writing immediately above the current line

At some point, you will want to leave vi and/or save your file. These commands detail how to leave and save in many combinations.
(Note that these are all commands. Vi must be in command mode for these to work.)

**save/quit commands:**
:wq     saves changes to file and quits the vi editor
:x     same as :wq
ZZ     same as :wq
:q!     quits without saving your changes (very useful)
:q     quits
:w     saves changes without quitting
:w name     same as :w but the file is saved to name
:..$w filea saves the rest of the file to filea

When you have written some text in a file, and want to edit it, you need to be able to move the cursor around in the file.

**moving commands:**

by character:
h     right
l     left

by word:
b     back one word
e     end of current word
w     next word

by line:
G to last line
j one line down
k one line up
:number go to line number
numberG same as :number
0 to beginning of current line
^ same as 0
$ to end of current line

general movement in file:
<ctrl>-b back a page
<ctrl>-u back half a page
<ctrl>-f forward a page
<ctrl>-d forward half a page
<ctrl>-i refresh screen
) forward to next sentence
( backward to previous sentence
% finds matching brace or parenthesis

The following editing commands are useful for editing characters, words, and lines as well as entire files.

special commands:
  u this is IMPORTANT.it undoes the last command you used
  U restores original line
  . (period) repeats last editing or insert command.
  :set number turns on line numbering
  :set nonumber turns off line numbering

editing commands:

for characters:
  r replaces the character the cursor is on with the next one typed.
  x deletes the character the cursor is on
  s substitutes a character with text and inserts text after the substitution until ESC is pressed.

for words:
  cw changes the word until ESC is pressed
  dw deletes word (or as much of it as is after the cursor.)

for lines:
  cc changes the line until ESC is pressed
  dd deletes line (also numberdd will delete number lines.)
  D deletes to end of line
  d$ same as D
  i<CR> splits the current line at the character the cursor is on. press ESC to end.
  J joins the current line to the next line
  :s/string1/string2 substitutes string2 for string1 in the current line
general editing for files:
R writes over old text.
d) deletes the rest of the sentence
d} deletes the rest of the paragraph
:e filea replaces current file with filea but saves it under the old filename
:x fileb reads in fileb at the cursor
%s/string1/string2 substitutes all occurrences of string1 with string2

Use these cut and paste commands to delete or move text in a file.
cutting and pasting commands.
cutting by word:
yw yanks a copy of the current word, or rest of the word, into a buffer
y3w yanks next 3 words from the cursor into a buffer
cutting by line:
ny yanks a copy of a line into a buffer
y$ yanks to end of line into a buffer
“a3yy} yanks 3 lines into a buffer called a
cutting by sentence/paragraph:
y) yanks a copy of the text to the end of sentence into a buffer
y} yanks to end of paragraph into buffer

p puts last text yanked or deleted (dd) after the cursor
P puts last text yanked or deleted before the cursor

Use these commands to find one or more of the occurrences of a word or a string in a file.
search commands:
/string searches file forwards for occurrence of string
n searches in the same direction (used with :s or /)
N searches in the opposite direction
?word searches for word backwards in the text before the cursor
I word searches for word forwards in the text before the cursor

Bridget Kromhout
Last modified 4-10-95