

# 0. Introduction

Why should you bother to learn Vim? Because you will be using it during your Practical Exam as File Transfers will be disabled during the Practical Exam. Also it is a powerful editor at the hands of a skilled programmer.

You do not need to memorize all these commands. You may print it out if you wish and use it as a reference sheet. Put it beside you when doing your programming. You will eventually find that you will use the reference sheet less and less. Practice makes better!

## 1. Vim Keys

When you startup Vim, you are in **Command Mode**. Normally, we press the "i" key to go into **Insert Mode**. You can type your Java program only when you are in **Insert Mode**.

Note that as usual, Vim is case-sensitive, i.e. "i" is different from "I". When typing "I", you should type "Shift-I".

### 1.1 Going from Command Mode to Insert Mode

There are many ways to go from **Command Mode** to **Insert Mode**:

- i Insert text before the cursor.
- I Insert text before the first non-blank in the line.
- o Begin a new line below the cursor and insert text.
- O Begin a new line above the cursor and insert text.
- a Append text after the cursor.
- A Append text at the end of the line.
- R Enter Replace mode: Each character you type replaces an existing character, starting with the character under the cursor.

### 1.2 Going from Insert Mode to Command Mode

This is easy. Simply press the <Esc> key.

### 1.3 Useful commands while typing in Insert Mode

- <Insert> Toggle between Insert and Replace mode.
- CTRL-N Search forward for next matching keyword. This keyword replaces the previous matching keyword.
- CTRL-P Search backwards for next matching keyword. This keyword replaces the previous matching keyword.
- CTRL-W Delete the word before the cursor.
- CTRL-U Delete all entered characters in the current line.
- CTRL-E Insert the character which is below the cursor.
- CTRL-Y Insert the character which is above the cursor.

## 2. Command Mode Commands

### 2.1 Visual Mode

v Start Visual mode per character.  
V Start Visual mode linewise.

When in Visual mode, use the arrow keys to move around to select your text (which is highlighted). Then choose one of the following actions:

y Yank (Copy) text.  
d Delete (Cut) text.  
c Delete (Cut) text and starts insert mode.  
~ Switch case of highlighted text.  
u Make highlighted text lowercase.  
U Make highlighted text uppercase.  
= Indent code.

If you have used Copy or Cut, you may use "p" or "P" to Paste your text as follows:

p Put (Paste) the text after the cursor.  
P Put (Paste) the text before the cursor.

### 2.2 Motion (Moving Around)

0 To the first character of the line.  
^ To the first non-blank character of the line.  
\$ To the end of the line.  
w One word forward.  
e Forward to the end of word.  
b One word backward.  
( One sentence forward.  
) One sentence backward.  
{ One paragraph forward.  
} One paragraph backward.  
H To line at top of screen.  
M To Middle line of screen.  
L To line at bottom of screen.  
CTRL-F Scroll window one page forward.  
CTRL-B Scroll window one page backwards.  
CTRL-E Scroll window one line downwards.  
CTRL-Y Scroll window one line upwards.  
f{char} To 1st occurrence of {char} to the right. The cursor is placed on {char}. E.g. "fg" moves the cursor to the 1st occurrence of the character "g" to the right.  
F{char} To 1st occurrence of {char} to the left. The cursor is placed on {char}.  
t{char} Till before 1st occurrence of {char} to the right. The cursor is placed on the character left of {char}.  
T{char} Till before 1st occurrence of {char} to the left. The cursor is placed on the character right of {char}.

gg	Goto the first line in the file, on the first non-blank character.
G	Goto the last line in the file, on the first non-blank character.
{count}G	Goto line {count} in the file, on the first non-blank character. E.g. "25G" goes to line 25. You may also go straight to line 25 from the Unix command prompt by typing: "vim +25 myfile.c"
%	Find the next item in this line after or under the cursor and jump to its match. Items can be: ([{}])                parenthesis or (curly/square) brackets /* */                start or end of C-style comment This is very useful to locate that missing closing ), }, ], or */.

## 2.3 Changing Text

All commands that start with ":" must end with "<Enter>".

r{char}	Replace the character under the cursor with {char}.
y{motion}	Yank (Copy) text that {motion} moves over. E.g. "yw" copies one word.
yy	Yank (Copy) one line.
Y	Yank (Copy) one line.
d{motion}	Delete (Cut) text that {motion} moves over. E.g. "dw" deletes one word. Another e.g. "dfn" deletes from the cursor to the first occurrence of "n" on the right (including the "n") in the line.
dd	Delete (Cut) one line.
D	Delete (Cut) the characters under the cursor until the end of the line. Synonym for "d\$".
c{motion}	Delete text that {motion} moves over and starts insert mode. E.g. "cw" deletes one word forward and starts insert mode.
cc	Delete one line and starts insert mode.
C	Delete the characters under the cursor until the end of the line and starts insert mode. Synonym for "c\$".
s	Delete one character under the cursor and starts insert mode.
S	Delete one line and starts insert mode.
x	Delete (Cut) one character under the cursor.
X	Delete (Cut) one character before the cursor.
~	Switch case of the character under the cursor and move the cursor to the right.
g~~	Switch case of the entire current line.
guu	Make the entire current line lowercase.
gUU	Make the entire current line uppercase.
J	Join the current line with the next line.
:r [name]	Insert the file [name] below the cursor. E.g. ":r sample.c" inserts the contents of the file "sample.c" below the cursor.

If you have used Copy or Cut, you may use "p" or "P" to Paste your text as follows:

p	Put (Paste) the text after the cursor.
P	Put (Paste) the text before the cursor.

Here are some useful key combinations:

xp	Exchange the order of two characters.
ddp	Exchange the order of two lines.
deep	Exchange the order of two words. The cursor must be placed in the blank space before the first word.

## 2.4 Repeating and Undoing Changes

.	Repeat last change.
u	Undo last change.
CTRL-R	Redo last change which was undone.

## 2.5 Doing Commands Multiple Times

Most commands above can be prefixed with a number to indicate the number of times to do the command. As you type the command, you should see the command appear at the bottom right hand corner of your screen. Here are some examples:

10dd	Delete (Cut) ten lines.
5yy	Yank (Copy) five lines.
72i*<Esc>	Insert "*" 72 times.
3fg	Moves the cursor to the 3rd occurrence of the character "g" to the right.

## 2.6 Marks

These commands are useful for jumping around quickly in your C program. You may set a mark at a certain position in your C program, then quickly jump back to it later.

m{a-zA-Z}	Set mark {a-zA-Z} at the current cursor position. It does not move the cursor, this is not a motion command. E.g. "md" sets mark "d" at the current cursor position. Since Vim is case sensitive, the mark "d" is different from the mark "D".
`{a-zA-Z}	Jump to the mark {a-zA-Z}. E.g. "`d" jumps to mark "d". Note that "`" is the backtick key, it is usually found on the top left hand portion of your keyboard.
``	Jump to the position before the last jump, or where the last "m" command was given (where the last mark was set).

## 2.7 Searching

All commands that start with "/" or "?" must end with "<Enter>".

{/pattern}	Search forward for the 1st occurrence of {pattern}. E.g. "/abc" searches forward for the 1st occurrence of the text "abc".
{?pattern}	Search backward for the 1st occurrence of {pattern}.
n	Repeat the latest "/" or "?" search.
N	Repeat the latest "/" or "?" search in the opposite direction.
*	Search forward for the 1st occurrence of the word under / nearest to the cursor.

#                   Search backward for the 1st occurrence of the word under /  
                    nearest to the cursor.

## 2.8 Find and Replace

All commands that start with ":" must end with "<Enter>".

`:%s/{pattern}/{string}`   Search the whole file and replace  
                          a match of {pattern} with {string}. Only the  
                          first occurrence of {pattern} in each line is  
                          replaced with {string}. E.g. "%s/abc/defg"

`:%s/{pattern}/{string}/g`   Search the whole file and replace  
                          a match of {pattern} with {string}. All  
                          occurrences of {pattern} in each line is  
                          replaced with {string}. E.g. "%s/abc/defg/g"

`:%s/{pattern}/{string}/gc`   Search the whole file and replace  
                          a match of {pattern} with {string}. All  
                          occurrences of {pattern} in each line is  
                          replaced with {string}. Each substitution is  
                          confirmed with a prompt. E.g. "%s/abc/defg/gc"

## 2.9 Saving and Quitting

All commands that start with ":" must end with "<Enter>".

`:w`     Write (save) the current file.  
`:wq`    Write (save) the current file and quit Vim.  
`:x`     Like ":wq", but write only when changes have been made.  
`ZZ`     Write current file, if modified, and quit Vim (same as ":x").  
`:q`     Quit Vim.  
`:q!`    Quit Vim without writing (saving).