Emacs

Definitions:

C-f	Means hold Ctrl and press f
M-p	Means hold alt (meta) and press p
ESC	Is the same as meta
C-x f	Means hold Ctrl and press x, release Ctrl and press f
C-x C-c	Means hold Ctrl and press x, keep Ctrl down and press c
C-M-e	Means hold Ctrl and alt and press e
C-S-+	Means hold Ctrl and shift and press +
Buffer	Is typically an opened file, but can also be a dir list
Region	Is the word for selection
Kill	Is the same as cut to clipboard
Yank	Is the same as paste

Load and Save

At the bottom of Emacs there is a small window – the mini buffer. It is used to display what's going on. You can also type in commands here. For instance, to open a file you press "C-x f", the cursor will move to the minibuffer and it will say something like "Find file c:/cetdev/version/home". Type in the filename and press return.

C-x f	Find file (to open)
C-x C-s	Save current buffer
C-x C-w	Save as

During find file:

SPACE	Show file list in some wind + complete
TAB	Auto-complete (based on current directory)
M-/	Auto-complete (based on all known text)

If you get stuck in the minibuffer:

C-g	QUIT – ends whatever you are doing (harmless)

Basic Navigation

C-f	Forward char (arrow right)
C-b	Backward char (arrow left)
M-f	Forward word
M-b	Backward word
C-n	Next line (arrow down)
C-p	Previous line (arrow up)
C-a	Beginning of line
C-e	End of line

C-F1	Remember position in code (same for all function keys)
F1	Return to position in code (same for all function keys)

M-v	Page up
C-v	Page down
M-z	Scroll up
C-z	Scroll down
Home	Beginning of buffer
End	End of buffer

Auto-complete

When you type any text, especially source code, you will type the same word over and over again. For speed and accuracy you will use emacs auto-completion. For instance, if you are going to use the class CScanStream, you type "CSc" and press M-/. Done.

Copy/Paste

C-space	Set mark (beginning or end of region)
M-w	Copy
C-M-w	Copy with colour
C-w	Cut/kill
C-y	Paste/yank
M-y	Pop and paste (use after C-y to access clipboard stack)

Undo/Redo

C-_	Undo (Ctrl-Shift--)
Spc C-_	Redo after undoing (Space Ctrl-Shift-)

More killing

C-d	Delete char
M-d	Delete (kill) word forwards
M-bcksp	Delete (kill) word backwards (typically M-f M-bcksp)
C-k	Kill rest of line (typically C-a C-k)

Search and replace

C-s	I-Search forward (incremental as you type)
C-s	Repeat last (move to next match)
C-r	Backward I-Search
C-M-s	Regexp I-Search
M-%	Query replace (Alt-Shift-5)
M-\	Auto search keyword

TAGS

In cmd	in c:\cetdev\version\home folder: make tags
C-c t	Build TAGS files, for current directory and subdirectories
C-c v	Visit TAGS, enter directory (c:\cetdev\version\home for all cm files)
M-s	TAGS search regexp
M-,	TAGS search next
M-q	TAGS-query-replace (i.e. marer+ press Enter + name + press Enter, then press 'Y'

More buffer handling

C-x k	Kill buffer (close it, will ask for save if necessary)
C-x b	Switch buffer
C-x l	Electric buffer list (list, select delete buffers)

Multiple windows

C-x 1	One window only
C-x 2	Split current to top and bottom
C-x 3	Split current to left and bottom
C-x 0	Close current window (remove split)
C-x 7	2 windows predefined config
C-x 8	3 windows predefined config
C-x 9	4 windows predefined config
C-c 7	2 windows low-red predefined config
C-.	Jump to next window
C-,	Jump to previous window
M- -	Shrink compilation window
M- =	Grow compilation window

Multiple frames

C-x 5 2	New frame
C-x C-c	Close frame or exit emacs (depending on frame)

Help me!

C-h k	Describe key (view what it binds to)
C-h f	Describe function
C-h v	Describe variable

M-x menu-bar-mode will enable the menu

CM Bindings

Editing

C-M-j	Insert a very large (file delimiter) comment
C-M-k	Insert a function header comment
C-]	Comment in/out line using//
C-M-c	Comment out region
M-;	Comment out region
C-M-b	To uncomment marked comment
Tab	Indent line
C-Tab	Indent line, go to next line.
Return	Indent, new line, indent
C-M-\	Indent region
C-o	Insert (unindented) line

Compiling

M-i	Compile (c++ compile only)
M-p	Run (c++ compile and run, cm just run)

Compiling in cm interact loop

C-M-u	Reload current
C-M-y	Reload all known
C-M-p	Run current
C-M-e	Exit
C-c C-m	Clean cm

To start cm, load any cm file and press M-p. From interact loop, reload buffers with C-M-u or C-M-y. If you get any strange errors, have to terminate the process manually or receive a c++ break point or and unhandled exception, there is a risk that cm will crash at boot at the next run. If it does make clean-cm

C-c C-x	Compile all code
C-c C-p	Compile all reachable code for current package
C-c C-v	Check for unreferenced local variables

Debug

M-n	Jump to next error
C-M-n	Jump to previous error
C-M-q	Quit (pop error level)
C-c s	Request stop (of infinite loop)
C-c C-c	Start c++ debugger

What you do is typically M-n, correct the error, reload with C-M-u and do pop error level with C-M-q. Then you should be back and running. The cm interact prompt “cm 1>” indicates the error level.

Navigation

M-i	Go to definition
C-M-e	Pop back from definition
C-c C-o	Overrides method
C-c C-i	Implements method
C-c C-s	Subclasses
C-c C-d	Parents
C-c C-w	Open class viewer

P4 Bindings

C-x p e	Check out current buffer
C-x p a	Add current buffer

P4 commands

What update actually does:

changes –q	(open changed files for edit)
p4 sync	(sync to server head revision)
changes –a	(open added/removed files for add/delete)

Running “changes –q” is always harmless. You should not run “changes” if you are not synchronized.

Typically when you are ready to commit your changes to the server, you do “update”, resolve the conflicts if any, and then submit from P4V or by “commit”.

Recommended that you use P4V visual client for commit.