Getting to know Prelude

Certainly the best way to understand how Prelude enhances the default Emacs experience is to peruse Prelude's source code (which is obviously written in Emacs Lisp). Understanding the code is not necessary of course. Prelude includes a prelude-mode minor Emacs mode which collects some of the additional functionality added by Prelude. It also adds an additional keymap that binds many of those extensions to keybindings.

https://github.com/bbatsov/prelude
https://github.com/dbushenko/emacs-config

CPP C++
https://www.amazon.com/dp/0321776402//ref=cm_sw_su_dp?tag=nethta-20

ww.newthinktank.com/2014/11/c-programming-tutorial/

Keymap

Global

Keybinding	Description
C-x \	align-regexp
C-+	Increase font size(text-scale-increase).
C--	Decrease font size(text-scale-decrease).
C-x O	Go back to previous window (the inverse of other-window (C-x o)).
C-^	Join two lines into one(crux-top-join-line).
C-x p	Start proced (manage processes from Emacs; works only in Linux).
C-x m	Start eshell.
C-x M-m	Start your default shell.
C-x C-m	Alias for M-x.
M-X	Like M-x but limited to commands that are relevant to the active major mode.

C-h A	Run apropos (search in all Emacs symbols).
C-h C-m	Display key bindings of current major mode and descriptions of every binding.
M-/	Run hippie-expand (a replacement for the default dabbrev-expand).
C-x C-b	Open ibuffer (a replacement for the default buffer-list).
F11	Make the window full screen.
F12	Toggle the Emacs menu bar.
C-x g	Open Magit's status buffer.
C-x M-g	Open Magit's popup of popups.
M-Z	Zap up to char.
C-=	Run expand-region (incremental text selection).
C-a	Run crux-move-beginning-of-line. Read this for details.

Prelude Mode

Keybinding	Description
C-c o	Open the currently visited file with an external program.
C-c i	Search for a symbol, only for buffers that contain code
C-c g	Search in Google for the thing under point (or an interactive query).
C-c G	Search in GitHub for the thing under point (or an interactive query).
C-c y	Search in YouTube for the thing under point (or an interactive query).
C-c U	Search in Duckduckgo for the thing under point (or an interactive query).
C-S-RET or Super-o	Insert an empty line above the current line and indent it properly.
S-RET or M-o	Insert an empty line and indent it properly (as in most IDEs).

C-S-up or M-S-up	Move the current line or region up.
C-S-down or M-S-down	Move the current line or region down.
C-c n	Fix indentation in buffer and strip whitespace.
C-c f	Open recently visited file.
C-M-\	Indent region (if selected) or the entire buffer.
C-c u	Open a new buffer containing the contents of URL.
C-c e	Eval a bit of Emacs Lisp code and replace it with its result.
C-c s	Swap two active windows.
C-c D	Delete current file and buffer.
C-c d	Duplicate the current line (or region).
C-c M-d	Duplicate and comment the current line (or region).
C-c r	Rename the current buffer and its visiting file if any.
C-c t	Open a terminal emulator (ansi-term).
C-c k	Kill all open buffers except the one you're currently in.
C-c TAB	Indent and copy region to clipboard
C-c I	Open user's init file.
C-c S	Open shell's init file.
C-c . +	Increment integer at point. Default is +1.
C-c . -	Decrement integer at point. Default is -1.
C-c . *	Multiply integer at point. Default is *2.
C-c . /	Divide integer at point. Default is /2.
C-c . \	Modulo integer at point. Default is modulo 2.
C-c . ^	Power to the integer at point. Default is ^2.
C-c . <	Left-shift integer at point. Default is 1 position to the left.

C-c . >	Right-shift integer at point. Default is 1 position to the right.
C-c . #	Convert integer at point to specified base. Default is 10.
C-c . %	Replace integer at point with another specified integer.
C-c . '	Perform arithmetic operations on integer at point. User specifies the operator.
Super-g	Toggle between God mode and non-God mode
Super-r	Recent files
Super-j	Join lines
Super-k	Kill whole line
Super-m m	Magit status
Super-m l	Magit log
Super-m f	Magit file log
Super-m b	Magit blame mode

Note: For various arithmetic operations, the prefix C-c . only needs to be pressed once for the first operation. For subsequent operations, only the appropriate operations (i.e. +, -, *, /... needs to be pressed).

macOS modifier keys

Prelude does not mess by default with the standard mapping of Command (to Super) and Option (to Meta).
If you want to swap them add this to your personal config:

```
(setq mac-command-modifier 'meta)
(setq mac-option-modifier 'super)
```

You can also temporarily swap them with C-c w (M-x prelude-swap-meta-and-super).
Note: I'd highly recommend to all macOS users to consider remapping Return to Control instead. That's an epic productivity boost and it's not as crazy as it sounds!

Projectile

Here's a list of functionality provided by Projectile:

Keybinding	Description
C-c p f	Display a list of all files in the project. With a prefix argument it will clear the cache first.

C-c p d	Display a list of all directories in the project. With a prefix argument it will clear the cache first.
C-c p T	Display a list of all test files(specs, features, etc) in the project.
C-c p s g	Run grep on the files in the project.
C-c p s s	Runs ag on the project. Requires the presence of ag.el.
M-- C-c p s g	Run grep on projectile-grep-default-files in the project.
C-c p b	Display a list of all project buffers currently open.
C-c p o	Runs multi-occur on all project buffers currently open.
C-c p r	Runs interactive query-replace on all files in the projects.
C-c p i	Invalidates the project cache (if existing).
C-c p R	Regenerates the projects TAGS file.
C-c p k	Kills all project buffers.
C-c p D	Opens the root of the project in dired.
C-c p e	Shows a list of recently visited project files.
C-c p a	Switch between files with the same name but different extensions.
C-c p c	Runs a standard compilation command for your type of project.
C-c p P	Runs a standard test command for your type of project.
C-c p z	Adds the currently visited to the cache.
C-c p p	Display a list of known projects you can switch to.

Prelude adds an extra keymap prefix S-p (S stands for Super), so you can use S-p instead of C-c p. By default on Windows keyboard Super is mapped to the Windows key and on macOS keyboards Super is mapped to the Command key.
If you ever forget any of Projectile's keybindings just do a:
C-c p C-h

Helm

Helm is setup according to this guide: A Package in a league of its own: Helm.

You can learn Helm usage and key bindings following the guide. C-c h is Prelude's default prefix key for Helm. If you don't remember any key binding, append C-h after C-c h for a list of key bindings in Helm.
If you love Helm and want to use Helm globally with enhanced helm-find-files, helm-buffer-lists..., you will have to also add (require 'prelude-helm-everywhere). When prelude-helm-everywhere is activated, Helm enables these global key bindings:

Key binding	**Description**
M-x	Run helm-M-x, an interactive version of M-x.
M-y	Run helm-show-kill-ring, shows the content of kill-ring.
C-x b	Run helm-mini, an interactive version of C-x b with more features.
C-x C-f	Run helm-find-files, an interactive version of find-file with more features.
C-h f	Run helm-apropos, an interactive version of apropos-command.
C-h r	Run helm-info-emacs, an interactive version of info-emacs-manual.
C-h C-l	Run helm-locate-library that can search for locations of any file loaded into Emacs.

This key binding is activated in shell-mode:

Key Binding	**Description**
C-c C-l	Run helm-comint-input-ring that shows shell history using Helm interface.

This key bindings is activated in eshell-mode:

Key Binding	**Description**
C-c C-l	Run helm-eshell-history that shows eshell history using Helm interface.

If you prefer Ido in everywhere, you should not add prelude-helm-everywhere, so you can use Helm along with Ido and Prelude's default commands.
You can always reactivate Helm with (prelude-global-helm-global-mode-on).

NOTICE: In helm-M-x, you have to pass prefix argument *AFTER* you run helm-M-x, because your prefix argument will be displayed in the modeline when in helm-M-x buffer. Passing prefix argument **BEFORE** =helm-M-x= **has no effect**.

Key-chords

Key-chords are available only when the prelude-key-chord module has been enabled.

Keybinding	Description
jj	Jump to the beginning of a word(avy-goto-word-1)
jk	Jump to a character(avy-goto-char)
jl	Jump to the beginning of a line(avy-goto-line)
JJ	Jump back to previous buffer(crux-switch-to-previous-buffer)
uu	View edits as a tree(undo-tree-visualize)
xx	Executed extended command(execute-extended-command)
yy	Browse the kill ring(browse-kill-ring)

Disabling key-chords

In some cases you may not want to have a key-chord that is defined by prelude, in which case you can disable the binding in your personal.el file by setting its command to nil. For example, to disable the jj key-chord add the following line:

```
(key-chord-define-global "jj" nil)
```

If you're an evil-mode user you'll probably do well to disable key-chord-mode altogether:

```
(key-chord-mode -1)
```