NeoCal Financial for iPad User Guide

Version 3.0

Published on September 4, 2017
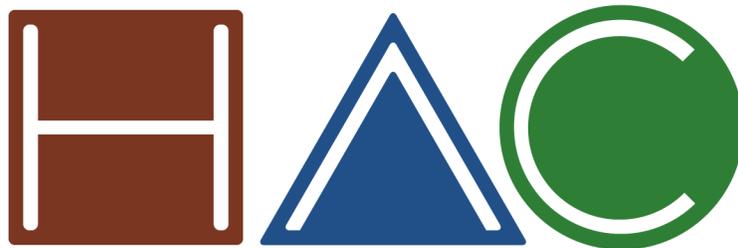
www.neocal.com

Table of Contents

Introduction

NeoCal™ is a calculator, and like many other calculators, you can start using it without reading a manual. This guide was written to provide a reference for its advanced features which may work similarly to, but not exactly the same as, the calculators you've already used.

NeoCal was designed to operate consistently and logically, with only a few exceptions added to match expectations of how other calculators work. If you come across something that doesn't make sense, please check this guide to get an understanding of NeoCal. The more complex applications are easy to use once you have a general idea of how NeoCal was designed.

Throughout this guide, examples are often provided in place of explanations, so it's important to work through them. Using the calculator while following the examples may improve your understanding of both the examples and the operation of the calculator.

Features

Here is a summary of some of NeoCal's advanced features:

- User interface optimized for the iPhone, iPod touch and iPad

- Efficient keyboard layouts for all screen sizes

- 12-digit display with thousands separators

- Display of fractions and mixed numbers

- Multi-level undo/redo support

- Keyboard macro recording and playback

- Choice of themes for different colors and dynamic keyboard characteristics

- Labeled results for ease of use and clarity

- 20 or 100 general-purpose user registers

- Choice of Simple, Precedence, and RPN input methods

NeoCal includes many functions:

- **Scientific:** trigonometric, hyperbolic, and logarithmic functions

- **Financial:** time value of money, amortization; discounted cash flow analysis for internal rate of return and net present value; interest conversion between nominal and effective

rates; margin and markup calculations

- **Statistical:** two variable statistical analysis including mean, weighted mean, standard deviation; linear regression forecasting, correlation coefficient; probability calculations

- **Date and Time:** date arithmetic; duration arithmetic and conversions including frames for digital video editing; moment conversions

- **Conversions:** length, area, volume, speed, weight, temperature; unit arithmetic

Getting Started

Keyboard Layouts

NeoCal contains far too many functions to place every button on the screen at once, and so several methods are provided for accessing these additional buttons on the advanced keyboard. The first method is to push the shift button (2nd) showing alternate functions for the buttons currently being displayed. Pressing the shift button a second time will restore the buttons to their non-shifted function.

Different functional groups can be selected by swiping or scrolling left and right to display one of the available keyboard pages such as *Lite*, *General*, *Financial*, *Statistical*, *Date & Time*, *Conversion 1 (Length, Area, Speed)*, or *Conversion 2 (Volume, Weight, Temp)*. Switching between keyboard pages during a calculation oe while entering a number does not have an effect on that calculation or the state of the calculator, and so calculations can involve operations from different functional groups.

The *Basic* and *Basic Plus* keyboard layouts are provided for doing simple calculations using larger, finger-friendly buttons.

The current keyboard layout for either portrait and landscape modes determines which keyboard pages are available. Choosing a different layout for each of these orientations allows for quickly switching between keyboard layouts by rotating the device.

NeoCal offers three more layouts to facilitate different usage scenarios on tablets. The first is the Landscape Split layout which is used in landscape mode when the keybaord does not occupy the entire width of the display. The Portrait Tall and Landscape Tall layouts allow for configuration when NeoCal is used in Slide Over or 1/3 Split View.

Selecting the input method changes the functions available in the main area of the keyboard and is discussed in the Input Methods section.

Input Methods

This calculator was designed to work equally well using either the *Simple*, *Precedence*, or *RPN* input methods. The simple input method is most familiar and allows entering calculations like [2] [+] [3] [=] to get the desired result of 5. A more complex example is 2 + 3 x 5 = 17, which can be entered two ways, [2] [+] [(] [3] [×] [5] [)] [=] or [3] [×] [5] [+] [2] [=]. The first sequence required the use of parentheses to instruct the calculator to multiply before adding the numbers (since the Simple input method does not conform to this algebraic convention). The

second sequence avoided the use of parentheses by re-ordering the operations to generate the correct result of 17.

The Precedence input method is an improvement over the simple input method in that binary operations are automatically performed before multiplication and division operations which are performed before addition and subtraction operations, reducing the need for parentheses. For example, the previous calculation can be entered without the parentheses as [2] [+] [3] [×] [5] [=] to obtain the result of 17. Additionally, [2] [×] [3] [yˣ] [2] [=] yields 18 since the power function is performed before the multiplication.

The benefits of the RPN, or Reverse Polish Notation, input method over the simple and precedence input methods become apparent when calculating complex expressions. Fewer mistakes are made since fewer keystrokes are required and the intermediate results are always displayed (providing feedback during the calculation, not just at the end). The order of the calculation is changed to avoid using parentheses, as in the previous example, so that the expression is evaluated from the inside working outward.

The general rule of thumb for RPN calculators is to enter the values, then the operation. Thus, [2] [+] [3] [=] becomes [2] [Enter] [3] [+]. The [Enter] key is used to separate numbers when entering more than one in succession. In the example, 2 + 3 x 5 = 17, the RPN method could be evaluated using either [2] [Enter] [3] [Enter] [5] [×] [+], or [3] [Enter] [5] [×] [2] [+]. Looking at these two sequences, we can isolate [3] [Enter] [5] [×] which always shows the result of 15. The two sequences then become [2] [Enter] 15 [+] and 15 [2] [+] which are more easily recognizable as 17.

Keystroke savings are realized in calculations that cannot be reordered to a form without parentheses. Let's look at the example 2 x 4 + 3 x 5 = 23. Using the simple input method we could calculate this expression as [2] [×] [4] [+] [(] [3] [×] [5] [)] [=], or without using parentheses, [2] [×] [4] [=] [MS] [3] [×] [5] [+] [MR] [=]. Using RPN, the sequence becomes [2] [Enter] [4] [×] [3] [Enter] [5] [×] [+]. Saving one keystroke is not that significant, but this is still a relatively simple calculation. One important distinction between the two input methods is demonstrated by the sub expression 2 x 4. Using the simple input method, this sometimes needs to be enclosed in parentheses (when not located at the beginning of the expression) and sometimes not. Using RPN, this sub expression is always entered the same way. This consistency also contributes to fewer mistakes being made.

One final point is that most of the calculations require the operand to be entered before the operation is executed. For example, calculating the square root of 2 is always entered [2] [√x] and the inverse of 10 is entered [1] [0] [1/x] regardless of the current input method. The RPN input method just extends this methodology to all of the calculator's functions.

The input method can be chosen in the General Settings view. If you'd like to learn more about how an RPN calculator works, check out the RPN Tutorial chapter of this guide. With all of the input methods, all of the calculator functions are available allowing you to work the way you're most comfortable.

The input method can be temporarily changed using the ⌷simp⌷, ⌷prec⌷, or ⌷rpn⌷ buttons. The clear operation restores the input method to the setting specified in the General Settings view.

Dynamic Keyboards

The keyboards are constructed dynamically to best take advantage of the available screen space while maintaining efficient operation based on user preferences which are specified by the current theme and optionally overridden in the Layout Settings view.

NeoCal will always select the largest font size for any given key. If the key is large enough, NeoCal can display both the primary and shifted functions at the same time.

The appearance and behavior of the keyboard can be controlled by a number of settings in the Layout settings view. These settings are described in the Shifted Functions section.

Formatting Numbers

The default number format is *Standard,* specifying numbers to be displayed in a natural format, with scientific notation used only when the number is too small or too large to fit on the display. *Fixed* format also displays the number in a natural format but only switches to scientific notation when the number is too large for the display (when the number is too small, it displays zero). Choosing the *Scientific* format will cause all numbers to be displayed using scientific notation. The *Engineering* format also uses scientific notation whereas the exponent is adjusted to a multiple of three.

Additionally, the number of digits that should be shown to the right of the decimal place can be specified. Selecting *All* causes all significant digits to be displayed (without trailing zeros, up to the limit of the display). Specifying a number for this field causes the display to show that number of digits past the decimal point including trailing zeros. If there are significant digits beyond what is being displayed, the last digit is rounded. This setting does not effect the accuracy of the calculations and is only used for displaying the results.

Pressing the ⌷show⌷ button will temporarily format the display showing all significant digits. Pressing ⌷show⌷ a second time, or performing any other operation, will return the format of the display to the specified setting. The *round* function, ⌷rnd⌷, will modify the internally stored

number to be exactly equal to the number shown on the display. This is useful when you don't want internally stored, but not displayed, digits to be a part of ongoing calculations.

In conjunction with setting the display format in the General Settings view, the ⌜std⌟, ⌜fix⌟, ⌜sci⌟, and ⌜eng⌟ buttons can be used to format the numbers in the display. Press one of these buttons to choose the format type followed by a single digit to specify the number of digits past the decimal point. Use the ⌜.⌟ button to specify all digits.

In addition to floating point numbers, NeoCal can also display mixed numbers and fractions. Changing the displayed representation using the ⌜float⌟, ⌜mix⌟, and ⌜frac⌟ is temporary, the clear operation returns the display to the format specified in General Settings. The displayed precision can be adjusted to reduce the values displayed in the numerator and denominator, helping to make the them more comprehendable.

NeoCal does not provide a special method for entering fractions, simply use the addition and division operations to separate the entries for the whole number, numerator and denominator.

Entering Numbers

Pressing any digit, the decimal point, or the exponent button causes the calculator to go into a special mode for entering numbers. The display format is not used; rather, the buttons pressed to enter the number are displayed. The rightmost character in the display gets erased when the ⌜bksp⌟ button is pressed.

When entering negative numbers, first enter the digits for the number, then press the ⌜+/−⌟ button. If you press the ⌜+/−⌟ button first, it will change the sign of the number currently being displayed, not the number you're about to enter.

When entering numbers in exponential format, first enter the mantissa, then press ⌜EEX⌟, and then enter the exponent. If the number is negative, press ⌜+/−⌟ *before* pressing ⌜EEX⌟. If the exponent is negative, press ⌜+/−⌟ *after* pressing ⌜EEX⌟.

If the ⌜bksp⌟ button is pressed while you're not entering a number, it will have the same effect as pressing ⌜CE⌟ or ⌜CLx⌟. If the ⌜bksp⌟ button is pressed after a prefix key, the prefix is cancelled.

Pressing any key to perform an operation will terminate the "entering a number" mode and the display will be formatted using the display settings.

Undo/Redo Support

The most recent operations can be reverted by swiping left on the display area of the main calculator view. An undo operation can be undone by swiping to the right. The entire state of the calculator, including stack and register values, are restored to the state before the previous operation was performed. Continue performing the Undo operation to continue reverting the calculator to previous states.

When another operation is performed after an undo operation, the redo information is lost. Also, the undo information is lost when the program is terminated.

The bksp button should be used to correct mistakes while entering numbers, as the undo operation will undo the entering number operation as a whole.

Macro Record and Playback

A macro stores a sequence of keystrokes or button presses that can be played back at a later time for simple automation of a common calculation. Only a single macro may be recorded or stored at a time.

- Press record to start recording a new macro. Any previously recorded macro is permanently deleted.

- Press pause to pause recording or playback of the macro. Press it a second time to resume recording or playback.

- Press stop to stop recording or playback of a macro.

- Press play to playback any previously recorded macro. It may be pressed again to resume a macro that has been paused for user input. The macro is played back using the same input method used to record it.

Using pause while recording a macro will pause the macro during playback allowing for input from the user. Keystrokes are not recorded while the macro is paused during recording, however, the user should provide similar keystrokes during playback before resuming the macro. Mistakes made during recording can be corrected using the undo operation.

Settings

There are many settings available in the Settings view to control and customize the operation and appearance of NeoCal. The settings can be opened by tapping on the right side of the calculator's display area or by tapping on the info button on the lower right corner if that is currently displayed.

General

All user settings that control the behavior of the calculator are located in the General Settings view which is accessed by selecting General on the main Settings view.

- **Input Method** specifies the input method or keystroke order, discussed in Input Methods . Use *Simple* for chained calculations, *Precedence* to perform binary operations before multiplication and division before addition and subtraction, and *RPN* for post-fix calculations without parentheses.

- The **RPN Stack Size** setting specifies the size of the stack used for RPN input method.

- The **Statistical Entry** setting specifies the input order when entering staticial data. The default method (y,x) matches the x and y registers for consistency with other calculator operations. This is often counter-intuitive, especially when not using the RPN input method, and so the (x,y) option is provided to allow entering data in the reverse order. The (x) setting tells NeoCal to only use a single value in the x register to collect data.

- The **Date Entry** setting specifies the format for entering dates. The default Y.MD value specifies dates will be entered in a Y.MMDD format while the other values provide alternate orderings such as D.MMYYYY and M.DDYYYY.

- Turning **100 Registers** on causes the user registers to be addressed using 2 digits, otherwise only the first 20 registers can be accessed.

- Enabling **Backspace Drops** switches the operation from clear x to drop when pressed and not entering a number.

Display

- The **Mode**, **Format** and **Precision** settings specify how numbers are displayed on the calculator's display. A detailed explanation is provided in the Formatting Numbers section.

- The **Radix Mark** setting specifies which character to use for the decimal point. Normally, this is determined by the system language settings, but can be overridden with period or comma options.

- The **Negative Parentheses** setting causes negative numbers to be indicated using parentheses instead of the normal minus sign.

- The **Long Date Format** setting controls whether dates will display in a longer format for easier reading. Both the short and long formats are determined by the locale setting for the device.

Keyboard

- **Shift Behavior** specifies the behavior of the shift button where *Sticky* causes the shift state to remain active whereas *Locking* remains active only after being pressed twice in a row.

- The **Sync Layouts** setting specifies whether NeoCal should try to change to the same keyboard page when the orientation is changed. If this value is *Never*, each orientation will remain on its own page as the device is rotated.

- The **Long Press** setting allows for an action to be performed when a long press is performed on a keyboard button. It is disabled by default.

Interface

- The **Key Clicks** setting specifies whether a sound is produced when a button on the keyboard is pressed.

- The **Keep Screen On** setting will prevent the device from sleeping while the main calculator screen is visible.

Appearance

NeoCal contains built-in themes that can be selected to alter the overall appearance of the main display and keyboard. Themes do not affect the operation of NeoCal in any way, but selecting a theme that is visually appealing can increase your efficiency when using NeoCal.

Generally, a theme specifies most of the visual aspects of the calculator including colors, margins, spacings, display lines, font size, border type, and dynamic keyboard settings. Some of

these characteristics can be specified or overridden in the Layout Settings view to further customize your interaction with NeoCal.

Settings with a default value are determined by the currently selected theme.

Display

- The **Negative Highlight** setting causes negative numbers to be displayed in an alternate color based on the current theme. A particular theme may not specify an alternate, causing these settings to not have an effect.

Shifted Functions

- The **Visibility** option determines whether the shifted functions should be shown along with the primary functions and, if so, where they should be displayed. *None* specifies they should never be shown even when there is plenty of room, *Above* specifies the shifted functions should be displayed above and outside the key boundaries, and *Within* specifies the shifted functions should be drawn on the face of the key (above the primary function). If there is not enough room to display all of the shifted functions, none of them will be displayed regardless of the current setting. To increase the likelihood that shifted functions will be shown, try reducing the font size and/or number of display lines, or hide the pager control.

- The **Keyboard** option specifies how the shifted keyboards should be displayed when the shifted functions are shown on the primary keyboard. The *Static* setting will not display a separate shifted keyboard, but continue to display the primary keyboard when the shift state is active. The *Swapped* setting switches the shifted function with the primary function indicating the lower (shifted) function is active and pressing the shift button again will access the primary function. The *Zoomed* setting will display keys with only the active (shifted) function visible. If the shifted functions are not displayed on the primary keyboard, then the shifted keyboards will be displayed with the equivalent of the Zoomed setting.

- The **Color** option specifies which color to use when drawing the shifted function on the primary keyboard. *Shift* indicates the shift color should be used, overridding the foreground color for that particular key. *Foreground* indicates the normal glyph color for that key should be used. *Background* causes the glyph to be drawn using the background color for that function. If the shifted functions are displayed within the key and this

setting is set to Background, the glyph will be drawn using the shift color. *Hybrid* will select an appropriate color with an emphasis on the clear buttons.

- The **Highlight** option specifies which color should be used for displaying the shifted (active) function on the shifted keyboard. The *Shift* setting specifies the glyph should be drawn using the shift color and the *Foreground* setting specifies the normal color used for drawing that function. The *Swapped* setting will draw the shifted function using the foreground color of the primary function, and draw the primary function using the shift color. *Hybrid* will select an appropriate color with an emphasis on the clear buttons.

Layouts

NeoCal layouts allow you to configure the display and keyboard to best suite the task at hand. For example, the **Portrait** orientation might be configured with a simpler keyboard, resulting in larger keys, that is more readily accessible for quick calculations while the **Landscape** layout might only be used occaisonally for tasks requiring more complex operations.

The **Landscape Split**, **Portrait Tall**, and **Landscape Tall** setting groups provide additional configuration points when the main calculator view does not occupy the entire screen such as when forms are shown or when multitasking on a tablet.

Everyone has different needs and uses a calculator in different ways, so NeoCal provides these layout settings to allow NeoCal to adapt to each user and situation.

- The **Layout** setting specifies which keyboard layout will be displayed for a particular calculator layout. The Plus and Tall layouts provide more keys per page and can be selected to better take advantage of larger screens.

- The **Show Forms** setting specifies whether the forms should be shown on the display next to the main calculator in landscape orientation. This setting can also be changed using the expand and collapse buttons on the toolbar.

- The **Swap Left/Right** setting controls whether the main part of the keyboard is displayed on the right or left side when in the full screen landscape orientation displayed.

Display

- **Lines** specifies the number of lines displayed on the main calculator screen. This setting can also be changed by swiping up or down on the display area.

- **Font Size** specifies the relative size of the font used to display results. A single line is normally displayed in the *Large* size, however, smaller fonts provide more space for the keyboard, resulting in larger keys that are easier to press.

Shifted Functions

- The **Visibility** option provides an override of the same Appearance setting for a particular layout. A *Default* value here defers to the selected value on the Appearance Settings view, which may itself defer to the value specified by the currently selected theme.

- If the shifted functions should be shown but there is not enough room to display them, NeoCal may change the primary function glyph to a smaller size to provide the room necessary to display the shifted functions. In this case, the **Minimum Size** option specifies the smallest acceptable size for the primary functions.

Interface

- The **Show Pager** setting allows the pager (normally shown below the keyboard) to be hidden, making more room to interact with the keyboard. The funtionality provided by the two buttons shown on either side the pager control can be accessed by tapping on the respective side of the calculator's display area.

Using Registers

An important concept of any calculator is the register. Each register can store a single value and is used for performing complex calculations or for remembering values for later use. These storage registers are further grouped into user (or numbered) and application specific registers.

Working Registers

NeoCal uses several registers for performing calculations with the most commonly used labeled *x*, *y*, and *last x*. The *x* register contains the number shown in the display, while the *y* register is paired with the *x* register when two values are required for a calculation or returned from a calculation. The *last x* register stores the value of the *x* register before the last numeric operation was performed.

As an example, let's add the numbers 12 and 15:

1. Pressing ⎯1⎯⎯2⎯ places the number 12 in the *x* register.

2. Pressing ⎯+⎯ copies the value stored in the *x* register into the *y* register.

3. Pressing ⎯1⎯⎯5⎯ stores the number 15 in the *x* register.

4. Pressing ⎯=⎯ adds the numbers in the *x* and *y* registers and places the result in the *x* register (which is shown on the display). The *y* register contains the value it contained before we started this calculation. The *last x* register contains the number 15.

Many keys are labeled with x (⎯1/x⎯) or x and y (⎯yˣ⎯) referring to the *x* and *y* registers. The general rule is to enter the *y* value first, then the *x* value. Also,

*   Pressing the ⎯swap⎯ button will swap the values stored in the *x* and *y* registers. This is often used when two values are returned from an operation and you want to view the additional value.

*   Pressing the ⎯last⎯ button recalls the value stored in the *last x* register into the *x* register. This allows a number to be used repeatedly without having to re-enter it.

The RPN input method also recognizes two more registers, *z* and *t*, which are discussed in RPN Tutorial.

The M Register

The *M* register is provided as a convenience when using the Simple or Precedence input method and is generally used to store intermediate values during the calculation. The following keys are used to access this register:

- [MC] is used to clear the *M* register, or set its value to zero.

- [MS] stores the value in *x* into the *M* register.

- [MR] recalls the *M* register by placing its value into *x*.

- [M+] adds *x* to the value already stored in the *M* register.

- [M–] subtracts *x* from the value currently stored in the *M* register.

The *MEM* indicator is displayed when the value of this register is nonzero.

User Registers

NeoCal can be configured for 20 or 100 storage registers for you to keep numbers beyond the current calculation. They are also useful for storing intermediate results for the current calculation or a series of calculations. If the 100 Registers options is not checked in the General Settings view, only twenty registers can be accessed and are numbered from 0 through 9 and .0 through .9 (this is the default setting). If the 100 Registers option is checked, then all 100 registers can be accessed by pressing two digits, 00 through 99, after the register operation. In either case, the registers are accessed using the these prefix buttons followed by the register number:

- [store] is used to store *x* into a specified register. The *STO* indicator is displayed.

- [recall] is used to recall a register's value into the *x* register. The *RCL* indicator is displayed.

- [xchg] exchanges the value stored in the *x* register with the value stored in the specified register. The *CHG* indicator is displayed.

The [bksp] button can be used to cancel the register prefix. Also, pressing a register prefix button twice in a row is a shortcut for displaying the User Registers form where tapping on one of the register values completes that operation.

For example (using 20 registers),

1. Pressing [1] [7] [store] [0] stores the number 17 in both the *x* register (by entering the number) and register 0.

2. Pressing ⬚2⬚ ⬚4⬚ [store] ⬚.⬚ ⬚5⬚ stores the number 25 in both the *x* register and register 15 (sometimes referred to as register .5).

3. Pressing [recall] ⬚0⬚ recalls the number 17 into the *x* register.

4. Pressing ⬚3⬚ ⬚3⬚ [xchg] ⬚.⬚ ⬚5⬚ stores 33 in register 15 and recalls 24 into the *x* register.

The same example using 100 registers,

1. Pressing ⬚1⬚ ⬚7⬚ [store] ⬚0⬚ ⬚0⬚ stores the number 17 in both the *x* register (by entering the number) and register 0.

2. Pressing ⬚2⬚ ⬚4⬚ [store] ⬚1⬚ ⬚5⬚ stores the number 25 in both the *x* register and register 15.

3. Pressing [recall] ⬚0⬚ ⬚0⬚ recalls the number 17 into the *x* register.

4. Pressing ⬚3⬚ ⬚3⬚ [xchg] ⬚1⬚ ⬚5⬚ stores 33 in register .5 and recalls 24 into the *x* register.

The store and recall operations can be used in conjunction with the add, subtract, multiply, and divide operations. Continuing the previous example,

5. Pressing ⬚3⬚ [store] ⬚+⬚ ⬚0⬚ ⬚0⬚ stores 3 into the *x* register and 20 (17 + 3) into register 0.

6. Pressing [recall] ⬚−⬚ ⬚1⬚ ⬚5⬚ recalls -30 (3 - 33) into the *x* register.

7. Pressing ⬚1⬚ ⬚1⬚ [store] ⬚÷⬚ ⬚1⬚ ⬚5⬚ stores 11 in the *x* register and -3 (-33 / 11) in register 15.

8. Pressing [recall] ⬚×⬚ ⬚0⬚ ⬚0⬚ recalls 220 (11 x 20) into the *x* register.

In addition to these general-purpose storage registers, there are ten financial registers and six statistical registers. They are discussed in the Financial Functions and Statistical Functions chapters of this guide, respectively.

User Registers View

The User Registers view displays the values of all 20 or 100 user registers.

Invoking the view by tapping in the display area, shows the [store], [recall], [xchg], and [swap] buttons. This allows multiple operations to be performed before the view is dismissed using the Done button. To perform an operation, simply tap on the register prefix button and then tap on the register value. The [swap] button is provided to facilitate moving values between multiple (more than two) user registers.

When the view is shown by pressing the same register prefix button twice, tap on the register value to complete the operation and dismiss the view.

Register Forms

Previously calculated values can be recalled from the stack, either on the calulator's main display or within the Stack form, by first pressing the ⌈recall⌉ button and then tapping on the number to be recalled.

The register operations, including arithmetic, can also be completed by tapping on the user register displayed alongside the keyboard in landscape mode, making the usage of these registers more efficient by providing context for a series of calculations.

Clearing The Registers

Storing zero into any register clears that register. The following keys are used to clear groups of registers:

- ⌈ C ⌉ or ⌈clear⌉ clears the *x*, *y*, *z*, *t*, and *last x* registers. This is also referred to as clearing the stack and should be performed before starting a new set of calculations. This operation also returns the current input method to the setting specified in the General Settings view.

- ⌈CE⌉ or ⌈CLx⌉ clears the *x* register. This should be used when an incorrect number has been entered and you'd like to start entering the number again.

- ⌈CA⌉ or ⌈CLA⌉ clears or resets all registers, clears the keyboard macro, and resets the calculator state. It does not modify any user settings.

- ⌈CLr⌉ clears the user, financial, and statistical registers.

- ⌈CLf⌉ clears the financial registers and displays the number of compounding periods per year. This function also clears any user registers that currently contain a cash flow entry, leaving all other user registers unmodified.

- ⌈CLΣ⌉ clears the statistical registers used to hold a summary of the entered data values.

- ⌈CLd⌉ resets the date conversion registers to their default values (except ⌈fps⌉) and displays the number of frames per second.

General Functions

It's always a good idea to clear the calculator before starting new calculations by pressing either the ⌷C⌷ or ⌷clear⌷ buttons. All of the functions in this section store the value of the x register into the *last x* register before performing the operation.

One-Number Functions

The following list of functions simply modify the value of the *x* register:

- ⌷1/x⌷ calculates the inverse of x, or x^-1.

- ⌷x²⌷ calculates the square of x, or x multiplied by x.

- ⌷√x⌷ calculates the square root of x.

- ⌷ip⌷ returns the integer portion of x.

- ⌷fp⌷ returns the fractional portion of x.

- ⌷abs⌷ returns the absolute value of x.

- ⌷ceil⌷ returns the smallest integral value not less than x.

- ⌷floor⌷ returns the largest integral value not greater than x.

Two-Number Functions

These functions require two numbers as input and store their result in the x register:

- ⌷+⌷ calculates the addition of x and y.

- ⌷-⌷ calculates the subtraction of x from y.

- ⌷×⌷ calculates the multiplication of x and y.

- ⌷÷⌷ calculates y divided by x.

- ⌷mod⌷ calculates the remainder of y divided by x.

- ⌷yˣ⌷ calculates y raised to the power of x.

- ⌷ˣ√y⌷ calculates the x root of y.

Percentages

Percent

Normally, the ⌈ % ⌋ function returns x divided by 100. However, it does perform a more useful function when the previous operation was addition or subtraction, then the number returned is the specified percentage of y. For example,

- Pressing ⌈ C ⌋⌈ 1 ⌋⌈ 5 ⌋⌈ % ⌋ displays the number 0.15.

- Pressing ⌈ C ⌋⌈ 5 ⌋⌈ 0 ⌋⌈ + ⌋⌈ 3 ⌋⌈ 0 ⌋⌈ % ⌋ displays the number 15 (since 30% of 50 is 15). The result after pressing ⌈ = ⌋ is 65.

In RPN mode, the result is always the specified percentage of y. The value of y is preserved so that subsequent operations can be performed without having to re-enter the base number.

Percent Change

The ⌈ Δ% ⌋ function calculates the percent difference between two numbers.

In RPN mode, the value of y is preserved.

Percent of Total

The ⌈ %T ⌋ function calculates the percentage that x is of y. For example,

- Pressing ⌈ 5 ⌋⌈ 0 ⌋⌈ %T ⌋⌈ 3 ⌋⌈ 0 ⌋ results in 60 (since 30 is equal to 60% of 50).

In RPN mode, the value of y is preserved.

Scientific Functions

Trigonometry

The angular mode buttons (deg, rad, and grad) specify whether angles are measured in degrees, radians, or grads. The *RAD* and *GRAD* indicators are displayed to indicate those modes, while the absence of these indicators implies degrees mode.

The following functions modify the number in the *x* register:

- sin calculates the sine of x.

- cos calculates the cosine of x.

- tan calculates the tangent of x.

- asin calculates the arc sine of x.

- acos calculates the arc cosine of x.

- atan calculates the arc tangent of x.

Pressing hyp before any of the above functions accesses the hyperbolic functions. The π button recalls the value of pi into the *x* register.

Logarithms

The following functions modify the number in the *x* register:

- ln calculates the natural log of x.

- e^x calculates the natural antilogarithm of x.

- log calculates the common log of x.

- 10^x calculates the common antilogarithm of x.

Statistical Functions

Statistical Registers

This application uses the statistical registers and performs one- and two-variable statistical calculations such as mean, standard deviation, and linear regression. The statistical registers are represented by the ⌐n⌐, ⌐Σx⌐, ⌐Σx²⌐, ⌐Σy⌐, ⌐Σy²⌐, and ⌐Σxy⌐ buttons. However, these registers are not modified using the ⌐store⌐ button, rather, data is accumulated into these registers by using:

- ⌐CLΣ⌐ to clear the statistical registers. This will also reset the Statistical Entry mode to that specified on the General Settings view.

- ⌐Σ+⌐ to accumulate a data point into the statistical registers. The data point can be either a single x value or an (x, y) pair. The number of accumulated data points is displayed. As with all NeoCal operations, the y value is entered before the x value.

- ⌐Σ-⌐ to remove a data point from the statistical registers. This is useful for correcting mistakes.

- ⌐(y,x)⌐ to temporarily change the entry mode to (y,x) where the x and y values correspond to the x and y registers.

- ⌐(x,y)⌐ to temporarily change the entry mode to (x,y) where the x value is entered before the y value.

- ⌐(x)⌐ to temporarily change the entry mode to (x) where the y value is not accumulated.

The accumulated statistics can be recalled by simply pressing these buttons:

- ⌐n⌐ recalls the number of data points accumulated.

- ⌐Σx⌐ recalls the summation of the x values.

- ⌐Σx²⌐ recalls the summation of the squares of the x values.

- ⌐Σy⌐ recalls the summation of the y values.

- ⌐Σy²⌐ recalls the summation of the squares of the y values.

- ⌐Σxy⌐ recalls the summation of the products of the x and y values.

- ⌐recall⌐ ⌐Σ+⌐ recalls the summation of the x values into the x register and the summation of the y values into the y register.

The Statistical form displays the current values of all the summary registers used by the statistical calculations.

Let's enter some data,

1. Press ⌷CL∑⌷ to clear the statistical registers.

2. Press ⌷1⌷ ⌷0⌷ ⌷Enter⌷ ⌷4⌷ ⌷0⌷ ⌷∑+⌷ to accumulate the first data point.

3. Press ⌷7⌷ ⌷Enter⌷ ⌷3⌷ ⌷5⌷ ⌷∑+⌷ to accumulate the second data point.

4. Press ⌷1⌷ ⌷5⌷ ⌷Enter⌷ ⌷7⌷ ⌷0⌷ ⌷∑+⌷ to accumulate the third data point.

5. Press ⌷3⌷ ⌷1⌷ ⌷Enter⌷ ⌷1⌷ ⌷0⌷ ⌷0⌷ ⌷∑+⌷ to accumulate the fourth data point.

6. Press ⌷∑x⌷ to recall the sum of the x values (245).

7. Press ⌷∑x²⌷ to recall the sum of the squares of the x values (17,725).

8. Press ⌷n⌷ to recall the number of accumulated data points.

Mean and Standard Deviation

The mean and standard deviation calculations are based upon the values accumulated in the statistical registers. These buttons are used to perform the calculations:

- ⌷x̄⌷ calculates and displays the mean (average) of the x values.

- ⌷ȳ⌷ calculates and displays the mean (average) of the y values.

- ⌷x̄w⌷ calculates and displays the weighted mean of the x values. The y values are used to determine the weight for the corresponding x value.

- ⌷s⌷ calculates the standard deviation of the x and y values and returns the results in the x and y registers, respectively.

- ⌷σ⌷ calculates the population standard deviation of the x and y values and returns the results in the x and y registers, respectively.

Continuing the previous example,

9. Press ⌷x̄⌷ to calculate the mean or average of the x values (61.25).

10. Press ⌷ȳ⌷ to calculate the mean or average of the y values (15.75).

11. Press ⌷x̄w⌷ to calculate the weighted mean of the x values (76.11).

12. Press ⌞s⌟ to calculate and display the standard deviation of the x values (30.10).

13. Press ⌞swap⌟ to display the standard deviation of the y values (10.69).

Linear Regression

Linear regression involves fitting a straight line through the set of two or more data points accumulated in the statistical registers. Linear estimation uses this line to calculate an estimate of either x or y, based on the other.

- ⌞x̂,r⌟ estimates x for a given y value. The correlation coefficient (r) is returned in the y register.

- ⌞ŷ,m⌟ estimates y for a given x value. The slope (m) of the line is returned in the y register. The y intercept of the line can be calculated by estimating y for x = 0.

Continuing the previous example,

14. Press ⌞1⌟⌞2⌟⌞x̂,r⌟ to estimate the x value (50.36) for the given y value.

15. Press ⌞swap⌟ to display the correlation coefficient (0.97).

16. Press ⌞0⌟⌞ŷ,m⌟ to estimate and display the y value (-5.34) for the given x value. This is the y-intercept of the fitted line.

17. Press ⌞swap⌟ to display the slope (0.34) of the fitted line.

Probability

The input for these calculations is restricted to non-negative integers.

- ⌞Cy,x⌟ calculates the number of possible sets of y different items taken x at a time.

- ⌞Py,x⌟ calculates the number of possible different arrangements of y different items taken x at a time.

- ⌞x!⌟ calculates the factorial of the displayed number.

- ⌞Γx⌟ calculates the gamma of the displayed number.

- ⌞rand⌟ returns a pseudo-random number between 0 and 1.

For example,

1. Press ⑤ ② Cy,x ⑤ to calculate the number of different five-card hands that can be dealt from a deck of cards (2,598,960).

2. Press ① ⓪ x! to calculate the factorial of 10 (3,628,800).

3. Press ① ⓪ Γx to calculate the gamma of 10 (362,880).

4. Press rand to generate a number between zero and one.

5. Press rand to generate a different number between zero and one.

Financial Functions

Financial Registers

NeoCal contains ten registers used specifically for the financial applications:

- ⌐cst⌐ is used for cost

- ⌐prc⌐ is used for price

- ⌐mar⌐ and ⌐mu⌐ are used for margin and markup

- ⌐P/YR⌐ is used for the number of periods or payments per year

- ⌐N⌐ is used for the total number of periods or payments

- ⌐I/YR⌐ and ⌐NOM%⌐ are used for the annual nominal interest rate

- ⌐PV⌐ is used for present value

- ⌐PMT⌐ is used for the periodic payment

- ⌐FV⌐ is used for future value

- ⌐EFF%⌐ is used for the annual effective rate

Just like the user registers, these registers are accessed using the ⌐store⌐ and ⌐recall⌐ buttons. However, the financial registers perform a calculation when their button is pressed without being preceded by ⌐store⌐ or ⌐recall⌐. As a convenience, pressing one of these buttons *while entering a number* will cause any pending operation to be performed and then store that number in the register.

For example,

- Pressing ⌐store⌐ ⌐FV⌐ will store the displayed number in the future value register.

- Pressing ⌐recall⌐ ⌐prc⌐ will display the value stored in the price register.

- Pressing ⌐7⌐ ⌐.⌐ ⌐5⌐ ⌐I/YR⌐ will store 7.5% in the annual interest rate register.

- Pressing ⌐4⌐ ⌐×⌐ ⌐1⌐ ⌐5⌐ ⌐N⌐ will store 60 in the N register.

- Pressing ⌐PV⌐ will calculate, store, and display the present value.

The financial registers and any user registers currently holding a cash flow entry are cleared by pressing the ⌐CLf⌐ button. As a reminder, the value of the ⌐P/YR⌐ register is displayed.

Margin and Markup

This application uses the ⌞cst⌟, ⌞prc⌟, ⌞mar⌟, and ⌞mu⌟ registers. Keeping in mind that the margin and markup registers are actually the same, any of these values can be calculated from the other two registers.

For example,

1. Pressing ⌞1⌟⌞0⌟⌞cst⌟ stores 10 in the cost register.

2. Pressing ⌞2⌟⌞0⌟⌞prc⌟ stores 20 in the price register.

3. Pressing ⌞mar⌟ calculates the margin to be 50 percent.

4. Pressing ⌞mu⌟ calculates the markup to be 100 percent.

5. Pressing ⌞3⌟⌞0⌟⌞mar⌟ stores 30 percent in the margin register.

6. Pressing ⌞prc⌟ calculates the price to be 14.29 (30% margin with a cost of 10, rounded to two decimal places).

7. Pressing ⌞recall⌟⌞cst⌟ displays the cost of 10.

8. Pressing ⌞recall⌟⌞mu⌟ displays the margin of 30 percent. Since the margin/markup register was last used to store the margin, recalling its value (using either button) will display the margin.

9. Pressing ⌞mu⌟ calculates the markup to be 42.86 percent.

Interest Conversion

This application uses the ⌞NOM%⌟, ⌞EFF%⌟, and ⌞P/YR⌟ registers to convert between the annual nominal and effective interest rates. The effective rate takes compounding into consideration based on the number of periods per year.

1. Pressing ⌞1⌟⌞2⌟⌞P/YR⌟ stores 12 in the periods per year register. This is equivalent to monthly compounding.

2. Pressing ⌞6⌟⌞NOM%⌟ stores 6 percent into the nominal interest rate register. This is equivalent to 0.5 percent per month.

3. Pressing ⌞EFF%⌟ calculates and stores 6.17 percent for the annual effective rate.

Since ⌞NOM%⌟ and ⌞I/YR⌟ share the same register, the annual effective rate can easily be converted for use by the time value of money application. In the United States, the annual effective rate is

equivalent to the annual percentage rate (APR).

Time Value of Money

This application can calculate any single value of the ⎡N⎤, ⎡I/YR⎤, ⎡PV⎤, ⎡PMT⎤, and ⎡FV⎤ registers based on the values of the other four registers. This is useful for compound interest calculations involving regular, uniform payments such as loans, mortgages, leases, and annuities. There must be at least one negative and one positive cash flow (⎡PV⎤, ⎡PMT⎤, and/or ⎡FV⎤). In addition,

- The value stored in the ⎡P/YR⎤ register is used to convert the annual nominal rate (⎡I/YR⎤) to the periodic rate (which is required for the calculations).

- ⎡×P⎤ stores the displayed number multiplied by the value in the ⎡P/YR⎤ register into the ⎡N⎤ register.

- ⎡÷P⎤ divides the displayed value by the value in the ⎡P/YR⎤ register.

- ⎡beg⎤ switches between *begin* and *end* modes and is used to specify whether the payments occur at the beginning or the end of the period. When *begin* mode is selected, the *BEG* indicator is displayed, otherwise *end* mode is in effect.

The Time Value of Money form displays the current values of all the registers used by the various TVM calculations.

For an example, let's consider a car loan. What would the payments be for a three-year loan of $16,500 with an APR of 9.5 percent?

1. If the calculator is in *begin* mode, press ⎡beg⎤ to set *end* mode.

2. Press ⎡CLf⎤ to clear the financial registers.

3. Press ⎡1⎤ ⎡2⎤ ⎡P/YR⎤ for monthly payments.

4. Press ⎡9⎤ ⎡.⎤ ⎡5⎤ ⎡EFF%⎤ to store the annual effective rate.

5. Press ⎡NOM%⎤ to calculate the annual nominal rate of 9.11 percent (which is stored in the ⎡I/YR⎤ register).

6. Press ⎡3⎤ ⎡×P⎤ to store 36 in the ⎡N⎤ register for the total number of payments.

7. Press ⎡1⎤ ⎡6⎤ ⎡5⎤ ⎡0⎤ ⎡0⎤ ⎡PV⎤ to store amount to be financed.

8. Press ⎡PMT⎤ to calculate -525.54 as the monthly payment amount. By convention, positive amounts designate money you receive and negative amounts designate money you pay.

As a comparison, let's calculate the payment for a four-year loan:

9. Press ⌨4 ⌨×P to store 48 for the total number of payments.

10. Press ⌨PMT to calculate -411.46 as the monthly payment amount.

Amortization

This application uses the ⌨I/YR ⌨PV ⌨PMT, and ⌨P/YR registers to calculate the amount of interest and principal for a range of payments. The remaining balance is also calculated.

- Pressing ⌨amort *while entering a number* will amortize for the range of payments starting with *y* and ending with *x* (inclusive).

- Pressing ⌨amort by itself will calculate the interest, principal, and balance for the range of payments currently entered (which defaults to the first payment, or range 1-1).

- Pressing ⌨anxt will amortize the next range of payments. For instance, if the current range of payments is 1-12 (the first year), then the next range is 13-24 (the second year).

- Pressing ⌨aprv will amortize the previous range of payments and is (currently only) available in the Amortization form.

- Pressing ⌨= after pressing ⌨amort or ⌨anxt will display the principal. Pressing ⌨= a second time will display the balance. As a convenience, pressing ⌨= a third time will calculate the amortization for the next range of payments.

- In RPN mode, press ⌨R↓ or ⌨drop to display the principal. Pressing ⌨R↓ a second time will display the balance. Press ⌨anxt to calculate the amortization for the next range of payments.

- Pressing ⌨store ⌨amort will store *y* and *x* as the starting and ending payments. (Pressing ⌨amort a second time will perform the calculation.)

- Pressing ⌨recall ⌨amort recalls the starting and ending payments into the *y* and *x* registers, respectively.

- ⌨beg switches between *begin* and *end* modes and is used to specify whether compounding occurs at the beginning or the end of the period. When *begin* mode is selected, the *BEG* indicator is displayed, otherwise *end* mode is in effect.

The amortization calculation is sensitive to rounding errors and may not match your financial institution's calculations exactly (different institutions use different rounding rules). NeoCal will

round the intermediate results based on the display setting for the number of digits to the right of the decimal place, therefore, this setting affects the results.

The Amortization form provides a convenient way of viewing all the registers used in the amortization calculation along with the calculated interest, principal, and balance.

Let's look at a mortgage example, $200,000 for 30 years at an annual nominal rate of 7 percent with payments and compounding occurring monthly.

1. Set the display to show two decimal places in the General Settings view.

2. If the calculator is in *begin* mode, press ⬚beg to set *end* mode.

3. Press ⬚CLf to clear the financial registers.

4. Press ⬚1 ⬚2 ⬚P/YR for monthly payments.

5. Press ⬚3 ⬚0 ⬚×P to store the loan duration in the ⬚N register.

6. Press ⬚7 ⬚I/YR to store the interest rate.

7. Press ⬚2 ⬚EEX ⬚5 ⬚PV to store the loan amount in the present value register.

8. Press ⬚PMT to calculate the monthly payment amount (-1330.60).

9. Press ⬚amort to amortize the first payment and display the interest amount (-1,166.67).

10. Press ⬚= (in RPN mode, press ⬚R↓) to display the principal for the first payment (-163.93).

11. Press ⬚= (in RPN mode, press ⬚R↓) to display the balance after the first payment (199,836.07).

12. Press ⬚= (in RPN mode, press ⬚anxt) to amortize the second payment and display the interest amount (-1,165.71). This process can be repeated through the term of the loan.

To calculate the amount of interest paid during the first year of the loan,

13. Press ⬚1 ⬚Enter ⬚1 ⬚2 ⬚amort to amortize the first twelve payments and display the interest amount (-13,935.65).

14. Press ⬚anxt to amortize the second year's payments and display the interest amount (-13,788.78).

At any time, you can press ⬚recall ⬚amort to recall the starting and ending payments for the range being amortized. (Use the ⬚swap button to display the starting payment.)

Cash Flow Analysis

This application uses the ⌷I/YR⌷ and ⌷P/YR⌷ registers. In addition, the user registers are used to store the cash flow entries. The net present value (⌷NPV⌷) function uses the ⌷I/YR⌷ register to calculate the present value of a series of cash flows and stores the result in the ⌷PV⌷ register. The internal rate of return (⌷IRR⌷) function calculates the annual nominal interest rate required to give a net present value of zero and stores the result in the ⌷I/YR⌷ register.

All cash flows must occur at regular intervals, but can be of differing amounts. Each cash flow entry consists of an amount and the number of times it occurs consecutively. Up to 99 cash flow entries can be stored after the initial cash flow. The amount of the initial cash flow is stored in register 0, the amount of the first cash flow is stored in register 1, the amount of the second cash flow is stored in register 2, and so on. When you enter a series of cash flows, you must account for every period, even those with a zero amount.

To enter the cash flows,

1. Press ⌷CLf⌷ to clear the financial registers and any cash flow entries.

2. Enter the number of periods per year in the ⌷P/YR⌷ register, if necessary.

3. Enter the initial cash flow amount and press ⌷CFj⌷.

4. Enter the next cash flow amount and press ⌷CFj⌷.

5. If this cash flow entry occurs more than once consecutively, enter the number of times it occurs and press ⌷Nj⌷.

6. Repeat steps 4 and 5 until all the cash flow entries have been entered.

To review all of the cash flow entries,

1. Press ⌷recall⌷ ⌷0⌷ (or ⌷recall⌷ ⌷0⌷ ⌷0⌷ if the 100 Registers option is enabled in the General Settings view) to display the initial cash flow amount.

2. Press ⌷recall⌷ ⌷CFj⌷ to display the amount of the next cash flow entry.

3. Press ⌷recall⌷ ⌷Nj⌷ to view the number of times that cash flow amount occurs.

4. Repeat steps 2 and 3 until all the cash flow entries have been reviewed.

The Cash Flow form can also be used to review the cash flow entries.

To review and edit the cash flow entries,

• Press ⌷recall⌷ followed by the number of the cash flow entry to display that amount.

- Press ⌊store⌋ followed by the number of the cash flow entry to store that amount.

- Pressing ⌊recall⌋ ⌊CFj⌋ displays the next cash flow amount.

- Pressing ⌊store⌋ ⌊CFj⌋ stores the next cash flow amount, but does not change the number of times the cash flow occurs.

- Pressing ⌊recall⌋ ⌊Nj⌋ displays the number of times the current cash flow occurs. The current cash flow entry is the entry whose amount was last stored or recalled.

- Pressing ⌊store⌋ ⌊Nj⌋ stores the number of times the current cash flow occurs.

- The cash flow amount can be changed by storing a new amount into the corresponding register, provided that the amount was initially stored using the ⌊CFj⌋ button.

- To add a new cash flow entry, first recall the last cash flow amount, then enter the new cash flow amount using the ⌊CFj⌋ button.

Let's suppose that you purchase $1000 worth of stock at the beginning of each of the next 12 months and then sell the stock two years from now for $16,000. What is the annual internal rate of return?

1. Press ⌊CLf⌋ to clear the registers.

2. Press ⌊1⌋⌊2⌋⌊P/YR⌋ to store the number of periods per year.

3. Press ⌊1⌋⌊0⌋⌊0⌋⌊0⌋⌊+/-⌋⌊CFj⌋ to store the initial cash flow.

4. Press ⌊CFj⌋ to store the first cash flow entry.

5. Press ⌊1⌋⌊1⌋⌊Nj⌋ to store the number of occurrences for this cash flow.

6. Press ⌊0⌋⌊CFj⌋ to store the second cash flow entry.

7. Press ⌊1⌋⌊2⌋⌊Nj⌋ to store the number of occurrences for this cash flow.

8. Press ⌊1⌋⌊6⌋⌊0⌋⌊0⌋⌊0⌋⌊CFj⌋ to store the last cash flow amount.

9. Press ⌊IRR⌋ to calculate the annual internal rate of return (18.71 percent) and store it in the ⌊I/YR⌋ register.

As another example, let's calculate the present value of twelve monthly payments of $1000 assuming 8 percent nominal annual interest rate and the first payment starting in one month from now:

1. Press ⌊CLf⌋ to clear the registers.

2. Press ⌊1⌋⌊2⌋⌊P/YR⌋ to store the number of periods per year.

3. Press ⌈8⌉⌈I/YR⌉ to store the interest rate.

4. Press ⌈0⌉⌈CFj⌉ to store the initial cash flow.

5. Press ⌈1⌉⌈0⌉⌈0⌉⌈0⌉⌈CFj⌉ to store the amount of the initial cash flow entry.

6. Press ⌈1⌉⌈2⌉⌈Nj⌉ to store the number of occurrences.

7. Press ⌈NPV⌉ to calculate the net present value (11,495.78) and store the result in the ⌈PV⌉ register.

Date and Time Functions

A *duration* represents a length of time, without starting or ending points. The duration units include years, months, weeks, days, hours, minutes, seconds, and frames. Frames are useful for digital video or film editing applications. Examples of durations include 2.5 years, 13 weeks, 8 days, and 75 minutes.

A *moment* represents a point in time and is expressed as a *date* (August 14, 2002), a *time* (5:15 pm), or a *datetime* (August 14, 2002 5:15 pm) value. NeoCal does not differentiate between timezones or daylight savings time adjustments, nor are leap seconds taken into consideration.

Durations

The duration conversion buttons either assign a duration unit (when the value in the x register is not already a duration) or convert a duration to another unit. The following buttons represent the duration conversions:

- `year` for years
- `mth` for months
- `week` for weeks
- `day` for days
- `hour` for hours
- `mnt` for minutes
- `sec` for seconds
- `fr` for frames

These conversions are determined by values stored in the following date conversion registers:

- `dpy` for the number of days per year (365.25)
- `dpm` for the number of days per month (30.4375)
- `dpw` for the number of days per week (7)
- `hpd` for the number of hours per day (24)
- `fps` for the number of frames per second (30)

Just like the user registers, these registers are accessed using the ⌈store⌉ and ⌈recall⌉ buttons. As a convenience, pressing one of these buttons *while entering a number* will cause any pending operation to be performed and then store that number in the register. Otherwise, pressing a date register button will recall it's value into the x register.

The ⌈CLd⌉ button resets the date conversion registers to their default values, except for the ⌈fps⌉ register whose value is displayed as a reminder. It is important to clear the date registers before performing a series of date calculations to ensure consistent results. (The ⌈CLd⌉ button also clears the date moment registers.)

The ⌈fpn⌉, or fractional part, button takes the fractional component of a duration and converts it to the next smaller unit.

For example,

1. Press ⌈CLd⌉ to reset the date registers.

2. Press ⌈1⌉ ⌈year⌉ to enter 1 year.

3. Press ⌈mth⌉ to convert to 12 months.

4. Press ⌈week⌉ to convert to 52.18 weeks.

5. Press ⌈day⌉ to convert to 365.25 days.

6. Press ⌈sec⌉ to convert to 31,557,600 seconds.

This example shows duration converions based on a forty-hour work week,

1. Press ⌈CLd⌉ to reset the date registers.

2. Press ⌈8⌉ ⌈hpd⌉ to store 8 in the hours per day register.

3. Press ⌈5⌉ ⌈dpw⌉ to store 5 in the days per week register.

4. Press ⌈1⌉ ⌈3⌉ ⌈4⌉ ⌈hour⌉ to enter 134 hours.

5. Press ⌈week⌉ to convert to 3.35 weeks.

6. Press ⌈fpn⌉ to calculate 1.75 days (0.35 weeks).

7. Press ⌈fpn⌉ to calculate 6 hours (0.75 days). This shows that 130 hours is the same as 3 weeks, 1 day, and 6 hours.

Duration arithmetic can also be performed, for example:

1. Press ⌈CLd⌉ to reset the date registers.

2. Press ⌷1⌷ ⌷hour⌷ ⌷+⌷ ⌷3⌷ ⌷0⌷ ⌷mnt⌷ ⌷=⌷ to add 1 hour and 30 minutes to generate the result of 1.5 hours.

3. Press ⌷1⌷ ⌷day⌷ ⌷+⌷ ⌷4⌷ ⌷8⌷ ⌷hour⌷ ⌷=⌷ to add 1 day and 48 hours to generate the result of 3 days.

4. Press ⌷3⌷ ⌷week⌷ ⌷-⌷ ⌷1⌷ ⌷2⌷ ⌷day⌷ ⌷=⌷ yields the result of 1.29 weeks. Pressing ⌷fpn⌷ shows this to be 1 week and 2 days.

In addition, there are three time conversion functions:

- ⌷→hms⌷ converts hours in decimal-fraction (H.h) format to minutes-second (H.MMSSs) format.

- ⌷→hr⌷ converts hours in minutes-second format to decimal-fraction format.

- ⌷→fr⌷ converts hours in minutes-second format to frames in decimal-fraction format.

For example,

1. Press ⌷2⌷ ⌷.⌷ ⌷5⌷ ⌷→hms⌷ to convert 2.5 hours to 2 hours 30 minutes (2.30000).

2. Press ⌷4⌷ ⌷.⌷ ⌷2⌷ ⌷3⌷ ⌷4⌷ ⌷8⌷ ⌷→hr⌷ to convert 4 hours 23 minutes 48 seconds to 4.3967 hours.

3. Press ⌷0⌷ ⌷.⌷ ⌷0⌷ ⌷1⌷ ⌷3⌷ ⌷0⌷ ⌷→fr⌷ to convert 1 minute 30 seconds to 2,700 frames (assuming 30 fps).

4. Press ⌷1⌷ ⌷0⌷ ⌷0⌷ ⌷0⌷ ⌷fr⌷ ⌷→hms⌷ to convert 1,000 frames to 33.3 seconds (0.00333).

Moments

Moments are stored internally as the number of seconds elapsed since January 1, 1904. Currently, only date and times through December 31, 2031 are supported. This is subject to change in a future version of NeoCal.

There are three date moment registers which are used for entering date and time values. They are:

- The ⌷date⌷ button is used for entering dates using a Y.MMDD format. For convenience, years less than 32 are assumed to be in the 21st century and years between 32 and 99 are assumed to be in the 20th century.

- The ⌷time⌷ button is used for entering times using a 24 hour H.MMSS format.

- The ⌊now⌋ button recalls the current date and time.

In addition to resetting the date conversion registers, the ⌊CLd⌋ also clears or resets the date moment registers to January 1, 1904 12:00 am.

Pressing the ⌊date⌋ or ⌊time⌋ buttons *while entering a number* will convert the number already in the x register. Otherwise, pressing one of these buttons will display a dialog for entering the desired value. In both cases, the value will be stored in that register for later recall.

For example,

1. Pressing ⌊2⌋ ⌊.⌋ ⌊0⌋ ⌊8⌋ ⌊2⌋ ⌊8⌋ ⌊date⌋ enters August 28, 2002 and stores it into the *date* register.

2. Pressing ⌊1⌋ ⌊4⌋ ⌊.⌋ ⌊3⌋ ⌊0⌋ ⌊time⌋ enters 2:30 pm and stores it into the *time* register.

3. Pressing ⌊now⌋ generates August 28, 2002 2:30 pm (the current date and time) and stores it into the *now* register.

4. Pressing ⌊date⌋ displays a dialog for entering a date and stores that date into the *date* register.

5. Pressing ⌊recall⌋ ⌊now⌋ recalls the last generated date and time value (August 28, 2002 2:30 pm) from the *now* register.

The Date Input mode can be specified in General Settings and specifies how dates are entered via the keyboard. The options are Y.MD (the default), D.MY, and M.DY. The number before the decimal point can be any number of digits, however, past the decimal point the day and month must be two digits and the year must be four digits. The full patterns would look like Y.MMDD, D.MMYYYY, and M.DDYYYY.

There are two moment conversion functions:

- ⌊dp⌋ for extracting the date part of a moment
- ⌊tp⌋ for extracting the time part of a moment

For example,

1. Pressing ⌊now⌋ generates August 28, 2002 2:45 pm.

2. Pressing ⌊dp⌋ returns August 28, 2002 as a date value.

3. Pressing ⌊recall⌋ ⌊now⌋ recalls August 28, 2002 2:45 pm into the x register.

4. Pressing ⌊tp⌋ returns 2:45 pm as a time value.

In addition, the following duration conversion functions will extract a portion of a moment:

- ⌈year⌉ for extracting the year from a date or datetime value

- ⌈mth⌉ for extracting the month from a date or datetime value

- ⌈day⌉ for extracting the day from a date or datetime value

- ⌈hour⌉ for extracting the hour from a time or datetime value

- ⌈mnt⌉ for extracting the minute from a time or datetime value

- ⌈sec⌉ for extracting the second from a time or datetime value

For example,

1. Pressing ⌈now⌉ generates August 28, 2002 3:00 pm.

2. Pressing ⌈year⌉ returns 2002.

3. Pressing ⌈recall⌉ ⌈now⌉ recalls August 28, 2002 3:00 pm into the x register.

4. Pressing ⌈hour⌉ returns 15.

Date Arithmetic

Date arithmetic involves the addition or subtraction of a moment with either a duration or another moment. The following combinations are supported:

- A moment plus or minus a duration results in a similar moment.

- A moment minus a similar moment results in a duration.

- A date plus a time results in a datetime.

Date arithmetic involving dates and datetimes assume duration units to be days, while calculations involving times assume hours. Durations of other units are converted to these units, however, the duration conversion registers are not used. Instead, these conversions always use the default values of these registers. Explicit conversions to days and hours must be performed to use the values stored in these registers.

For example,

1. Press ⌈date⌉ and enter today's date (August 28, 2002).

2. Press ⌈+⌉ ⌈9⌉ ⌈0⌉ ⌈=⌉ to calculate 90 days into the future (November 26, 2002).

3. Press ⎡ - ⎤⎡ 1 ⎤⎡ 8 ⎤⎡ 0 ⎤⎡ = ⎤ to calculate 90 days into the past (May 30, 2002).

4. Press ⎡recall⎤⎡date⎤ to recall today's date (August 28, 2002).

5. Press ⎡ + ⎤⎡ 3 ⎤⎡week⎤⎡ = ⎤ to calculate 3 weeks into the future (September 18, 2002).

To calculate the day of year:

6. Press ⎡recall⎤⎡date⎤ to recall today's date (August 28, 2002).

7. Press ⎡ - ⎤⎡ 1 ⎤⎡ . ⎤⎡ 1 ⎤⎡ 2 ⎤⎡ 3 ⎤⎡ 1 ⎤⎡date⎤⎡ = ⎤ to calculate the number of days since December 31, 2001 (240).

To time an event:

1. Press ⎡now⎤ to generate the starting datetime (August 28, 2002 9:15 am).

2. At the end of the event, press ⎡recall⎤⎡now⎤ to recall the starting datetime.

3. Press ⎡ - ⎤⎡now⎤ to generate the ending datetime (August 28, 2002 2:30 pm).

4. Press ⎡ = ⎤⎡+/-⎤ to calculate the length of time (5.25 hours).

To construct a datetime value, add separate date and time values:

1. Press ⎡ 2 ⎤⎡ . ⎤⎡ 0 ⎤⎡ 8 ⎤⎡ 2 ⎤⎡ 8 ⎤⎡date⎤ to enter the date part (August 28, 2002).

2. Press ⎡ + ⎤⎡ 1 ⎤⎡ 5 ⎤⎡ . ⎤⎡ 1 ⎤⎡ 5 ⎤⎡time⎤ to enter the time part (3:15 pm).

3. Press ⎡ = ⎤ to combine the datetime value (August 28, 2002 3:15 pm).

Conversion Functions

The conversions are divided into six types: length, area, volume, speed, weight, and temperature. The conversion buttons are pressed to either assign a unit (when the number does not have a unit, or the unit is not of the same type) or convert to another unit of the same type. In either case, the units assigned to the number are the same as the button.

The conversions are separated into two function groups with the Length, Area, and Speed conversions in the first group and the Volume, Weight (or Mass), and Temperature conversions in the second group.

In addition, duration conversions are available in the Date & Time function group and discussed in the Date and Time Functions section.

Length Conversions

The following buttons represent the length conversions:

- [in] for inches

- [ft] for feet

- [yd] for yards

- [mi] for miles

- [nmi] for nautical miles

- [mm] for millimeters

- [cm] for centimeters

- [m] for meters

- [km] for kilometers

For example,

1. Press [1] [2] [in] to enter 12 inches.

2. Press [ft] to convert to 1 foot.

3. Press [m] to convert to 0.30 meters.

Area Conversions

The following buttons represent the area conversions:

- [in²] for square inches

- [ft²] for square feet

- [yd²] for square yards

- [mi²] for square miles

- [acre] for acres

- [mm²] for square millimeters

- [cm²] for square centimeters

- [m²] for square meters

- [km²] for square kilometers

For example,

1. Press [1] [m²] to enter one square meter.

2. Press [ft²] to convert to 10.76 square feet.

3. Press [in²] to convert to 1500 square inches.

4. Press [m²] to convert back to 1 square meter.

Volume Conversions

The following buttons represent the volume conversions:

- [tsp] for teaspoons

- [tbsp] for tablespoons

- [fl.oz] for fluid ounces

- [cup] for cups

- [pint] for pints

- [qt] for quarts

- [gal] for gallons

- <u>fl.oz uk</u> for fluid ounces UK

- <u>gal uk</u> for gallons UK

- <u>ml</u> for milliliters

- <u>liter</u> for liters

- <u>in³</u> for cubic inches

- <u>ft³</u> for cubic feet

- <u>yd³</u> for cubic yards

- <u>cm³</u> for cubic centimeters

- <u>m³</u> for cubic meters

For example,

1. Press <u>1</u> <u>gal</u> to enter one gallon.

2. Press <u>liter</u> to convert to 3.79 liters.

Speed Conversions

The following buttons represent the speed conversions:

- <u>ft/s</u> for feet per second

- <u>mi/h</u> for miles per hour

- <u>knot</u> for knots

- <u>cm/s</u> for centimeters per second

- <u>m/s</u> for meters per second

- <u>km/h</u> for kilometers per hour

For example,

1. Press <u>5</u> <u>5</u> <u>mi/h</u> to enter 55 mph.

2. Press <u>km/h</u> to convert to 88.51 km/h.

Weight Conversions

The following buttons represent the weight conversions:

- $\boxed{\text{oz}}$ for ounces

- $\boxed{\text{lb}}$ for pounds

- $\boxed{\text{ton}}$ for tons

- $\boxed{\text{mg}}$ for milligrams

- $\boxed{\text{g}}$ for grams

- $\boxed{\text{kg}}$ for kilograms

For example,

1. Press $\boxed{1}$ $\boxed{\text{lb}}$ to enter one pound.

2. Press $\boxed{\text{oz}}$ to convert to 16 ounces.

3. Press $\boxed{\text{kg}}$ to convert to 0.45 kilograms.

Temperature Conversions

The following buttons represent the temperature conversions:

- $\boxed{\text{°F}}$ for degrees Fahrenheit

- $\boxed{\text{°C}}$ for degrees Celsius

For example,

1. Press $\boxed{7}$ $\boxed{5}$ $\boxed{\text{°F}}$ to enter 75 degrees Fahrenheit.

2. Press $\boxed{\text{°C}}$ to convert to 23.89 degrees Celsius.

Unit Arithmetic

Addition and subtraction can be performed on measurements with the same type of unit (except temperature) or with constants. If the units are not the same type, the result will not have a unit assigned to it. If necessary, the second operand's units will be converted to the first operand's units.

For example,

1. Press ⬚3⬚ ⬚ft⬚ ⬚+⬚ ⬚1⬚ ⬚5⬚ ⬚in⬚ ⬚=⬚ to add fifteen inches to three feet. The result is 4.25 feet, or 4 feet 3 inches.

2. Press ⬚1⬚ ⬚gal⬚ ⬚+⬚ ⬚3⬚ ⬚liter⬚ ⬚=⬚ to add one gallon and 3 liters. The result is 1.79 gallons.

Multiplication can be performed on two lengths to generate an area and an area divided by a length will produce a length. The ⬚x²⬚ and ⬚√x⬚ functions can also be used for these calculations.

Multiplication between an area and a length yields a volume, while a volume divided by an area or length yields the appropriate unit.

For example,

1. Press ⬚3⬚ ⬚ft⬚ ⬚×⬚ ⬚4⬚ ⬚ft⬚ ⬚=⬚ to generate the result of 12 square feet.

2. Press ⬚÷⬚ ⬚6⬚ ⬚ft⬚ ⬚=⬚ for the result of 2 feet.

3. Press ⬚3⬚ ⬚ft⬚ ⬚x²⬚ to generate 9 square feet.

Units are preserved through the following numeric functions: ⬚+/−⬚, ⬚ip⬚, ⬚fp⬚, ⬚abs⬚, and ⬚rnd⬚.

RPN Tutorial

Automatic Memory Stack

The RPN input method is based upon working with numbers stored in the automatic memory stack, which consists of the x, y, z, and t registers. The stack *lift* and *drop* operations are performed automatically and are key to understanding its use. Imagine the registers stacked on top of each other with t (on the top) above z, which is above y, which is above x. Both the lift and drop operations cause values to be shifted to an adjacent register, but shift in opposite directions.

The stack lift operation is performed to make room for a new value in the x register: the value in the t register is thrown away, the t register gets the z value, the z register gets the y value, and the y register gets the x value. This allows the new x value to be stored while preserving the original x value (in the y register).

The stack drop operation is performed when an operation requiring two operands is executed: the t register remains unaffected, the z register gets the t value, the y register gets the z value, and the x register gets the y value. The result of the calculation is then stored in the x register, effectively combining (or consuming) the original x and y values. Notice that the value in the t register is propagated downward. This is useful for operations involving multiple uses of a constant value.

Calculator operations can be categorized as either lifting the stack, dropping the stack, or neither lifting nor dropping the stack. Also, some operations disable stack lift, meaning that if the next operation normally lifts the stack, then the stack won't be lifted. Or, if the next operation lifts the stack twice, the stack will only be lifted once.

Let's look at a simple example to calculate the square of the sum of two numbers,

1. Press ⎣4⎦ to enter the number 4. Entering a number is a stack lifting operation, so the previously displayed value is moved into the y register.

2. Press ⎣Enter⎦ to separate the two numbers. This operation lifts the stack and then disables stack lift (see the next section for details on the enter button). The number 4 is now stored in both the x and y registers.

3. Press ⎣5⎦ to enter the second number. Since stack lift was disabled, 5 is simply stored in the x register.

4. Press ⎣+⎦ to add the numbers stored in the x and y registers, drop the stack, and then store the result in the x register. The number that was displayed before we started this example

is now stored in the y register.

5. Press x^2 to calculate the square of the displayed number. This operation does not lift nor drop the stack; only the value in the x register is modified.

By default the stack contains four values, but the stack size can be increased in the General Settings view.

Stack Manipulation

There are eight buttons used only for directly manipulating the stack: Enter, R↓, R↑, swap, drop, fill, rot, and over. As noted in the previous example, the Enter button lifts the stack and then disables lift for the next operation. This is primarily used for entering two numbers in succession, but can also be used for propagating a single value into all four stack registers by pressing it three times.

The R↓ and R↑ buttons rotate the values in the stack registers while preserving all values. The R↓ button is similar to stack drop except the x value is moved into the t register. The R↑ button is similar to stack lift except the t value is moved into the x register. Pressing either of these buttons four times in a row will result in the register values stored in their original locations.

The drop button performs a stack drop while preserving the value in the t register. The fill button copies the value in the x register into all the registers in the stack.

The swap button exchanges the values in the x and y registers.

The rot operation rotates the bottom three (x, y, and z) values upward. The over button pushes the y value onto the stack, moving all values upward and duplicating the y value.

The Stack form displays all the registers in the automatic stack. This form also includes the rot, over, drop, and swap stack manipulation buttons.

Frequently Asked Questions

General FAQ

Why can't I find a button mentioned in the documentation?

Many buttons are only accessible by first pressing the shift button (2nd) which is located above the 7 button.

Why do the I/YR and IRR calculations return 'error'?

These solutions are found using a 'solve' algorithm which sometimes fails to find the correct solution. The current value in the I/YR register is used as a starting point or estimate, and if this value is too different from the answer, the algorithm might fail to find a correct solution and return 'error'. Try providing an estimate or guess by storing an approximate value in the I/YR register before performing the calculation.

Why do some buttons appear in more than one location?

This is done strictly for convenience, providing easy access to the buttons you need, when you need them. The duplicated buttons operate the same regardless of their location.

Revision History

Version 1.0

- Initial version for iPhone and iPod touch

- Created the Lite, Scientific, and Financial editions that include a subset of the functionality available in the traditional Advanced edition

- Implemented first class support for landscape keyboard layouts

- Optimized the keyboard layouts for better spacial recognition when switching between portrait and landscape orientations

- Created a new set of themes

- Included the User Guide within the application

- Added the Statistical Entry and Date Entry settings along with appropriate keys and status indicators

- Added the Negative Parentheses setting

- Replaced the SPC key with Enter for non-RPN input methods

- Renamed several keys to better match user expectations and reduce possible future conflicts.

- Added Future Value on the Amortization form

- Removed the prompt for the Clear All operation (undo restores the calculator state)

Version 1.0.1

- Improved the readability of the button graphics in the User Guide

Version 2.0

- Initial version for iPad, released as a universal app

- Changed key clicks setting to on/off

- Increased minimum version requirement to iOS 4

- Fixed keyboard drawing bug when shift mode was set to sticky

• Fixed cash flow form displaying entries when none were entered

Version 2.0.1

• Added support for retina display

• Relabeled keys to swap and xchg

• Fixed crash when scrolling past end of keyboard in landscape mode

Version 2.0.2

• Rearranged keys on Lite page for better locality and consistency

• Modified to prevent multiple popups

Version 2.0.3

• Added support for retina 4" display and iOS 6

• Improved keyboard visualization, feedback, and performance

• Improved look of existing themes and added four new ones

• Implemented new display gestures

• Added calculation of odd roots of negative numbers

• Corrected calculation when entering consecutive binary operations in infix modes, ignoring all but the last

• Modified the Stack form to display entire stack when not in RPN mode

• Added Radix Mark setting to support locale-specific number formatting

• Added Keep Screen On setting

• Improved the readability of the embedded User Guide

• Fixed the temporary nature of changing input modes from the keyboard

• Fixed crash when changing some keyboard layout settings

• Increased minimum required version to iOS 5

Version 3.0

• Added support and updated design for iOS 10

- Added support for Split View and Slide Over

- Added full screen landscape layout

- Added ability to display fractions and mixed numbers

- Added recall stack operation

- Modified the D.MY and M.DY date input modes to expect 4-digit years

- Fixed bug with User Registers form when accessing registers addressed with a decimal point

- Renamed keys to store, recall and last

- Added Long Press setting to access shifted functions

- Added Plus and Tall keyboard layouts for larger displays

- Added Portrait Tall and Landscape Tall layout settings

- Simplified appearance settings

- Added Carbon and Air themes

- Redesigned application icon

- Improved User Guide navigation

- Changed default number of registers to 20

- Changed website to neocal.com and Twitter to @neocalapp

- Increased requirement to iOS 9 or later