

NeoCal Lite for iPad User Guide

Version 2.0.3

Published on July 23, 2013



www.hudren.com

Copyright © 2000-2013 Hudren Andromeda Connection. All rights reserved.

NeoCal and the HAC logo are trademarks of Hudren Andromeda Connection. Other company and product names may be trademarks or registered trademarks of their respective holders.

Table of Contents

INTRODUCTION	1
Features	1
GETTING STARTED	2
Keyboard Layouts	2
Input Methods	2
Dynamic Keyboards	4
Formatting Numbers	4
Entering Numbers	4
Undo/Redo Support	5
SETTINGS	6
General	6
<i>Display</i>	6
<i>Keyboard</i>	7
Theme	7
Layout	7
<i>Display</i>	7
<i>Interface</i>	8
USING REGISTERS	9
Working Registers	9
The M Register	10
User Registers	10
User Registers View	11
Clearing The Registers	12
GENERAL FUNCTIONS	13
No Argument Functions	13
One-Number Functions	13
Two-Number Functions	13

Percentages	13
<i>Percent</i>	14
<i>Percent Change</i>	14
<i>Percent of Total</i>	14
SCIENTIFIC FUNCTIONS	15
Trigonometry	15
Logarithms	15
CONVERSION FUNCTIONS	16
Length Conversions	16
Area Conversions	16
Volume Conversions	17
Speed Conversions	18
Weight Conversions	18
Temperature Conversions	19
Unit Arithmetic	19
RPN TUTORIAL	21
Automatic Memory Stack	21
Stack Manipulation	22
FREQUENTLY ASKED QUESTIONS	23
General FAQ	23
REVISION HISTORY	24
Version 1.0	24
Version 2.0	24

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 and iPod touch
- Efficient keyboard layouts
- 12-digit display with thousands separators
- Multi-level undo/redo support
- Choice of themes for different colors and dynamic keyboard characteristics
- Labeled results for ease of use and clarity
- 100 general-purpose user registers
- Choice of Simple, Precedence, and RPN input methods

NeoCal includes many functions:

- **Scientific:** trigonometric, hyperbolic, and logarithmic functions
- **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 scrolling left and right to display one of the *Lite*, *Conversion 1 (Length, Area, Speed)*, or *Conversion 2 (Volume, Weight, Temp)* keyboard pages. Switching between keyboard pages during a calculation 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* keyboard layout is 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.

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 $2 + 3 \times 5 = 17$, 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 $2 + 3 \times 5 =$ to obtain the result of 17. Additionally, $2 \times 3^{y^x} 2 =$ 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 $2 \text{ Enter } 3 +$. The **Enter** key is used to separate numbers when entering more than one in succession. In the example, $2 + 3 \times 5 = 17$, the RPN method could be evaluated using either $2 \text{ Enter } 3 \text{ Enter } 5 \times +$, or $3 \text{ Enter } 5 \times 2 +$. Looking at these two sequences, we can isolate $3 \text{ Enter } 5 \times$ which always shows the result of 15. The two sequences then become $2 \text{ Enter } 15 +$ 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 $2 \times 4 + 3 \times 5 = 23$. Using the simple input method we could calculate this expression as $2 \times 4 + (3 \times 5) =$, or without using parentheses, $2 \times 4 = \text{MS } 3 \times 5 + \text{MR } =$. Using RPN, the sequence becomes $2 \text{ Enter } 4 \times 3 \text{ Enter } 5 \times +$. 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 2×4 . 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 $2 \sqrt{x}$ 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.

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.

Several different sets of glyphs are available for drawing the keys and NeoCal will always select the largest glyph for any given key. In other words, the bigger the key, the bigger the writing on it and the smaller the key, the smaller the writing. If the key is large enough, NeoCal can display both the primary and shifted functions at the same time.

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.

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 **bksd**

button is pressed.

When entering negative numbers, first enter the digits for the number, then press the $\boxed{+/-}$ button. If you press the $\boxed{+/-}$ 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 \boxed{EE} , and then enter the exponent. If the number is negative, press $\boxed{+/-}$ *before* pressing \boxed{EE} . If the exponent is negative, press $\boxed{+/-}$ *after* pressing \boxed{EE} .

If the $\boxed{\text{bksp}}$ button is pressed while you're not entering a number, it will have the same effect as pressing $\boxed{\text{CE}}$ or $\boxed{\text{CLx}}$. If the $\boxed{\text{bksp}}$ 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 $\boxed{\text{bksp}}$ button should be used to correct mistakes while entering numbers, as the undo operation will undo the entering number operation as a whole.

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 stack operations.
- 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 **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 **Negative Highlight** and **Integer Highlight** settings cause these numbers to be displayed in an alternate color based on the current theme.

- 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.

Theme

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.

Layout

Separate keyboard layouts may be selected for the **Portrait** or **Landscape** orientations. Choosing either of these on the Settings view displays the settings for that orientation. Settings with a default value are determined by the currently selected theme.

- The **Layout** setting specifies which keyboard layout will be displayed.

Display

- **Lines** specifies the number of lines displayed on the main calculator screen.
- **Font Size** specifies the relative size of the font used to display results. Smaller fonts provide more space for the keyboard, resulting in larger keys that are easier to press.

Interface

- The **Show Pager** setting allows the pager (normally shown below the keyboard) to be hidden, making more room to draw and interact with the keyboard. The functionality provided by the two buttons shown along side the pager 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 $\boxed{1}$ $\boxed{2}$ places the number 12 in the x register.
2. Pressing $\boxed{+}$ copies the value stored in the x register into the y register.
3. Pressing $\boxed{1}$ $\boxed{5}$ stores the number 15 in the x register.
4. Pressing $\boxed{=}$ 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 ($\boxed{1/x}$) or x and y ($\boxed{y^x}$) referring to the x and y registers. The general rule is to enter the y value first, then the x value. Also,

- Pressing the $\boxed{\text{swap}}$ 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 $\boxed{\text{LSTx}}$ 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 has 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. If the 100 Registers option is checked, then all 100 registers can be accessed by pressing two digits 00 through 99. In either case, the registers are accessed using the these prefix buttons followed by the register number:

- **STO** is used to store x into a specified register. The *STO* indicator is displayed.
- **RCL** 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** **STO** **0** stores the number 17 in both the x register (by entering the number) and register 0.

- Pressing $\boxed{2} \boxed{4} \boxed{\text{STO}} \boxed{.} \boxed{5}$ stores the number 25 in both the x register and register 15 (sometimes referred to as register .5).
- Pressing $\boxed{\text{RCL}} \boxed{0}$ recalls the number 17 into the x register.
- Pressing $\boxed{3} \boxed{3} \boxed{\text{xchg}} \boxed{.} \boxed{5}$ stores 33 in register 15 and recalls 24 into the x register.

The same example using 100 registers,

- Pressing $\boxed{1} \boxed{7} \boxed{\text{STO}} \boxed{0} \boxed{0}$ stores the number 17 in both the x register (by entering the number) and register 0.
- Pressing $\boxed{2} \boxed{4} \boxed{\text{STO}} \boxed{1} \boxed{5}$ stores the number 25 in both the x register and register 15.
- Pressing $\boxed{\text{RCL}} \boxed{0} \boxed{0}$ recalls the number 17 into the x register.
- Pressing $\boxed{3} \boxed{3} \boxed{\text{xchg}} \boxed{1} \boxed{5}$ 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,

- Pressing $\boxed{3} \boxed{\text{STO}} \boxed{+} \boxed{0} \boxed{0}$ stores 3 into the x register and 20 (17 + 3) into register 0.
- Pressing $\boxed{\text{RCL}} \boxed{-} \boxed{1} \boxed{5}$ recalls -30 (3 - 33) into the x register.
- Pressing $\boxed{1} \boxed{1} \boxed{\text{STO}} \boxed{+} \boxed{1} \boxed{5}$ stores 11 in the x register and -3 (-33 / 11) in register 15.
- Pressing $\boxed{\text{RCL}} \boxed{\times} \boxed{0} \boxed{0}$ recalls 220 (11 x 20) into the x register.

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 $\boxed{\text{STO}}$, $\boxed{\text{RCL}}$, $\boxed{\text{xchg}}$, and $\boxed{\text{swap}}$ 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 $\boxed{\text{swap}}$ 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.

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, and resets the calculator state. It does not modify any user settings.
- **CLr** clears the user registers.

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.

No Argument Functions

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

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 .
- **x!** calculates the factorial of the displayed number.

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 .
- **y^x** calculates y raised to the power of x .
- **^xy** calculates the x root of y .

Percentages

Percent

Normally, the $\boxed{\%}$ 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 $\boxed{C} \boxed{1} \boxed{5} \boxed{\%}$ displays the number 0.15.
- Pressing $\boxed{C} \boxed{5} \boxed{0} \boxed{+} \boxed{3} \boxed{0} \boxed{\%}$ displays the number 15 (since 30% of 50 is 15).
The result after pressing $\boxed{=}$ 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 $\boxed{\Delta\%}$ function calculates the percent difference between two numbers.

In RPN mode, the value of y is preserved.

Percent of Total

The $\boxed{\%T}$ function calculates the percentage that x is of y . For example,

- Pressing $\boxed{5} \boxed{0} \boxed{\%T} \boxed{3} \boxed{0}$ 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*.

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.

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:

- in^2 for square inches
- ft^2 for square feet
- yd^2 for square yards
- mi^2 for square miles
- acre for acres
- mm^2 for square millimeters
- cm^2 for square centimeters
- m^2 for square meters
- km^2 for square kilometers

For example,

1. Press 1 m^2 to enter one square meter.
2. Press ft^2 to convert to 10.76 square feet.
3. Press in^2 to convert to 1500 square inches.
4. Press m^2 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
- fl.oz uk for fluid ounces UK

- **gal uk** for gallons UK
- **ml** for milliliters
- **liter** for liters
- **in³** for cubic inches
- **ft³** for cubic feet
- **yd³** for cubic yards
- **cm³** for cubic centimeters
- **m³** for cubic meters

For example,

1. Press **1** **gal** to enter one gallon.
2. Press **liter** to convert to 3.79 liters.

Speed Conversions

The following buttons represent the speed conversions:

- **ft/s** for feet per second
- **mi/h** for miles per hour
- **knot** for knots
- **cm/s** for centimeters per second
- **m/s** for meters per second
- **km/h** for kilometers per hour

For example,

1. Press **5** **5** **mi/h** to enter 55 mph.
2. Press **km/h** to convert to 88.51 km/h.

Weight Conversions

The following buttons represent the weight conversions:

- for ounces
- for pounds
- for tons
- for milligrams
- for grams
- for kilograms

For example,

1. Press to enter one pound.
2. Press to convert to 16 ounces.
3. Press to convert to 0.45 kilograms.

Temperature Conversions

The following buttons represent the temperature conversions:

- for degrees Fahrenheit
- for degrees Celsius

For example,

1. Press to enter 75 degrees Fahrenheit.
2. Press 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 to add fifteen inches to three feet. The result is 4.25

feet, or 4 feet 3 inches.

2. Press $\boxed{1}$ $\boxed{\text{gal}}$ $\boxed{+}$ $\boxed{3}$ $\boxed{\text{liter}}$ $\boxed{=}$ 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 $\boxed{x^2}$ and $\boxed{\sqrt{x}}$ 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 $\boxed{3}$ $\boxed{\text{ft}}$ $\boxed{\times}$ $\boxed{4}$ $\boxed{\text{ft}}$ $\boxed{=}$ to generate the result of 12 square feet.
2. Press $\boxed{\div}$ $\boxed{6}$ $\boxed{\text{ft}}$ $\boxed{=}$ for the result of 2 feet.
3. Press $\boxed{3}$ $\boxed{\text{ft}}$ $\boxed{x^2}$ to generate 9 square feet.

Units are preserved through the following numeric functions: $\boxed{+/-}$, $\boxed{\text{ip}}$, $\boxed{\text{fp}}$, $\boxed{\text{abs}}$, and $\boxed{\text{rnd}}$.

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 , $\text{R}\downarrow$, $\text{R}\uparrow$, 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 $\text{R}\downarrow$ and $\text{R}\uparrow$ buttons rotate the values in the stack registers while preserving all values. The $\text{R}\downarrow$ button is similar to stack drop except the x value is moved into the t register. The $\text{R}\uparrow$ 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.

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 Negative Parentheses setting
- Added the Negative Highlight 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.
- 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

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

- Center tap shows list of keyboard pages
- Swipe horizontally for undo / redo
- Long press number for clipboard operations
- 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 crash when changing some keyboard layout settings
- Increased minimum required version to iOS 5