

EECS 2311

YU-82 CALCULATOR

User Manual

(Group 3):
Brooks Dulla
Andrew Ferreira
Tedi Papajorgji
Dan Sheng
Yari Yousefian

Lassonde School of Engineering
York University

Table of Contents:

Introduction.....	3
Post-Fix Notation.....	3
Overview.....	4
Entering Numbers.....	5
Addition.....	5
Subtraction.....	5
Multiplication.....	6
Division.....	6
Negation.....	6
Decimal.....	7
Factorial.....	7
Radian Conversion.....	7
Sine.....	8
Cosine.....	8
Pi.....	8
Clear.....	9
Undo.....	9
Mode.....	10
Select.....	10

Introduction:

Thank you for purchasing the YU-82 Calculator. Designed and created by Group 3, this calculator is constantly being upgraded having new features added. This calculator is designed to be in post-fix notation (or Reverse Polish Notation (RPN)), quite different from the commonly used infix notation.

Post-Fix Notation:

Post-fix notation removes the need of an equals sign and generally operates by adding elements into a stack for calculation. The simplest example of post-fix is the addition of '1' and '2'. This is done in post-fix by:

1 2 +

The '1' is placed into the stack first since was entered first, with '2' following behind. The entering of an operator pops two operands from the stack (in this case '1' and '2') and applies the operation. The final value is then placed back in to the stack. For addition and multiplication, either operand can apply the operation to the other. However, for subtraction and division, the post-fix notation formally has the second operand removed from the stack apply the operation onto the first operand.

(Post-fix) 20 5 - is equivalent to 20 - 5 (Infix)

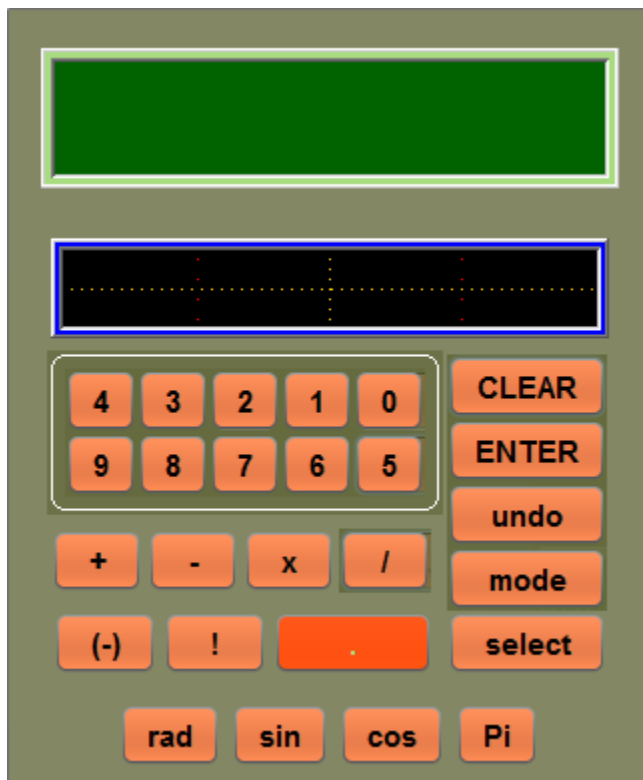
Overview:

The calculator itself consists of two displays, operands, operators, functions, constants, and operations to modify operand inputs.

The green display at the very top contains the current operands entered, as well as the history of button presses. The blue display below has both a vertical and horizontal grid, used to graph any sine or cosine function if a sine or cosine button at the very bottom is pressed. Otherwise, it is empty.

The drawn box within contains all 10 Arabic numerals as operands and when pressed, are displayed in the green display.

Below that is a row of operators that do addition, subtraction, multiplication, and division. Further below is a operand negation button, factorial operator, and a decimal point for operands.



At the very bottom row, there is a degree-to-radian converter button, sine and cosine operations, and a button to input the constant pi.

Lastly, on the very right it contains the calculator's functions such as clearing the display and history, entering a value to store in the stack, undoing previous operand entries, modifying history display, and selecting certain indexes in the history/stack.

The YU-82 Calculator

Entering Numbers:

To enter numbers in to the calculator, simply use the number pad to enter the number and store it in the stack by pressing **ENTER**.

Multiple numbers can be entered and stored in the stack.

Ex. Entering 4 and 2:



Addition:

Post-fix requires two operands to be entered before addition can occur. Otherwise, the calculator will only add zero if only one operand is entered, or throw an error if no values are stored. To add, simply **ENTER** two operands and press **+**.

Ex. For $1 + 2$:



Subtraction:

Subtraction is similar to addition in terms of actions the calculator will undergo if the amount of operands stored is less than two. As for subtraction, it was explained in Post-fix Notation (page XXX) that if two numbers are stored the number entered first will subtract the second. To subtract, simply **ENTER** two operands and press **-**.

Ex. For $3 - 1$:



Multiplication:

To multiply, **ENTER** two operands and press **X**.

Ex. For 3 X 2:



Division:

Like subtraction, the first operand entered/stored will be divided by the second operand. To divide, **ENTER** two operands and press **/**.

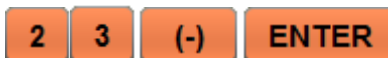
Ex. For 4/2:



Negation:

To store a negative value, the **(-)** button is used to turn the number in the display negative. Pressing the button again turns the number back to positive.

Ex. Entering -23:



Decimal:

To store a decimal value, the `'.'` button is pressed to insert a decimal point. It is important to note that the calculator only stores a maximum of 2 decimal points. Anything more and it will be rounded off.

Ex. Entering 43.26:



Factorial:

The factorial button `'!'` is a unary operator which means it is an operation with only one operand needed. However, this calculator does not support the factorial of decimals and the factorial of negative integers is not possible.

Ex. Finding 5!:



Radian Conversion:

The **"rad"** button converts the current displayed value from degrees into radians. However, it should be noted that the sine and cosine functions **DO NOT** take radians, but instead degrees.

Ex. Convert 180° to radians:



Sine:

The "**sin**" button takes the currently displayed value **A** as an angle and displays the value for $\sin(A)$, as well as the graph for it in the blue display. The blue display also has a small circle pointing to the location of the value. The angle for sin can only be an integer and the circle will only display the location if the angle is between -360° and 360° .

To find $\sin(A)$, enter the operand, press **sin**, then **ENTER**.

Ex. $\sin(45^\circ)$:



Cosine:

The "**cos**" button has the same behaviour as the "**sin**" button, but just displays the value for $\cos(A)$.

To find $\cos(A)$, enter the operand, press **cos**, then **ENTER**.

Ex. $\cos(270^\circ)$:



Pi:

The "**pi**" button is simply a constant to use for calculations. Pressing this button just displays pi to 2 decimal points in the green display. However, it only displays it and **ENTER** must be pushed to store it in the stack.

Ex. Circumference of circle, radius = 4 cm:



Clear:

The "**CLEAR**" button clears the display and entire stack history. This is commonly used when a different set of calculations will be entered.

Ex. Finding $1+4$ then $6-2$:



Undo:

The "**undo**" button undoes the last pressed button if entering a number and displays an error if nothing left to undo. If pressing "**undo**" after performing an operation, then it effectively performs the function of the "**CLEAR**" button. However, it displays

"0 | Start new computation".

Ex. Entering 152 but making a mistake and entering 14 first:



Mode:

The "**mode**" button simplifies the history operations in the display. Since the history displays the previous operations in infix notation, pressing "**mode**" simplifies an operation with two operands to the answer.

Ex. Adding $11 + 4$ and pressing **mode**:

1 1 ENTER 4 ENTER +

Gives

15 | 11+4

mode

Gives

15 | 15

Select:

The "**select**" button is perhaps the most versatile feature in the calculator. It allows the user to select one or more elements from the list of entered values in the stack and perform operations on them. These operations include the basic four operators (**+**, **-**, **X**, **/**) as well as the **undo** button.

The **select** can only perform the undo function on the displayed list of elements in the stack, not previous operations. To select an element, the index number of the element is needed. The right-most number in the display will have the index number **1**, with subsequent elements to the left having **2**, **3**, etc.

To select a single element, enter the element's index number and press **select**. The display will be empty and therefore allows the user to perform an operation. **Undo** will delete it from the stack while the four operators will perform the operation with the selected element and the left-most operand in the display.

Ex. Removing the 2nd element:

1 ENTER 2 ENTER 3 ENTER 4 ENTER

Gives

4 | 4,3,2,1

2 select undo

Gives

4 | 4,3,1

Ex. Copying the 1st element as the last element:

1 ENTER 2 ENTER 3 ENTER 4 ENTER

Gives

4 | 4,3,2,1

1 select +

Gives

1 | 1,4,3,2,1

Ex. Removing elements 1 to 3:

1 ENTER 2 ENTER 3 ENTER 4 ENTER

Gives

4 | 4,3,2,1

1 . 3 select undo

The above is important to have **LOW_INDEX.HIGH_INDEX**.
The reverse order will not work.

Gives

4 | 4

Ex. Subtracting elements 2 to 3:

1 ENTER 2 ENTER 3 ENTER 4 ENTER

Gives

4 | 4,3,2,1

2 . 3 select -

Gives

-1 | -1,4,3,2,1

This means the **2** subtracts **3** from itself to get **-1**, which is place into the front. The same idea applies to any other operation (**+**, **X**, **/**).

If the user selects 3 or more elements and does subtraction, then the smallest index element subtracts the other two elements from itself.

As for division, the smallest index element is divided by the one larger index's element. The quotient is then divided by the next (or in the case of 3 selected elements, the largest index's element).