IT WORKSHOP PROJECT

SCIENTIFIC CALCULATOR
(PYTHON)

BY:

BT21CSE079 MAMULLA PRANAY
BT21CSE077 ROBIN K. BANSAL
BT21CSE106 ANANDAPU. CHAITANYA KUMAR

ABOUT THE PROJECT

This project is about creating a scientific calculator using python. This calculator can perform math operations such as : Addition, Subtraction, Division, Multiplication, Power, Modulus, Pi, Exponential, Trigonometry, Factorial and Logarithmic operations.

Working of the CODE :

The code is a simple python program using the **Graphical User Interface** (GUI). it is a form of interface that allows the user to interact with the electronic devices through graphical icons and audio indicators. This makes the display more presentable and correct functioning of software applications.

Libraries used :

Tkinter : It is a Python binding to the Tk Gui toolkit. Tkinter is a standard Python interface to the Tk GUI. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter is light weight and relatively easy to use compared to other frameworks.

Modules used :

Time module : Python has time module that deals with time related tasks
Math module : Importing this will allow you to use the math functions
Os module : The os module provides a portable way to use operating system functionalities like reading or writing a file.

The whole calculations are done when the user clicks the **‘=’**

eval() : the eval() method parses the expression passed to this method and runs the python expression within the python. **eval() Parameters :**

The eval() function takes three parameters:

- **expression -** the string parsed and evaluated as a python expression
- **globals(**optional **) -** a dictionary
- **locals(**optional **) -** a mapping object. Dictionary is the standard and commonly used mapping type in python.

The eval() method returns the result evaluated from the expression.

Creating the application:

root=Tk() :
The root window is created. The root window is a main application window in our program. It consists of a title bar and borders. It must be created before any other widgets.
It is basically the window that we insert everything , in simple words it is the frame of the application.

root .geometry(“axb+c+d”) :
The geometry method sets a size for the window and positions it on the screen. The first two parameters are the width and height of the window. The last two parameters are the x and y coordinates of the screen.

root.title(‘xyz’) :
This gives the title to the application window.

root.mainloop :
Finally we enter the mainloop. The mainloop receives events from the window system and dispatches them to the application widgets. It is terminated when we click on the close button of the title bar or the quit() method.

Menu :

To give some information and get feedback we have created menu that has 1.

About

- Information - selecting this option gives a brief overview of the calculator.
- Feedback - this option allows the user to share their experience with the app.

2. Help
   This gives a message about the errors.
3. Exit
   End the program.

Buttons :

The Button widget is used to add buttons in the application. These can display text or images that convey the purpose of the buttons. We can attach a function or a method to a button which is called automatically when you click the button.

This repeated for 5 rows, changing the value of row from 1 to 5.

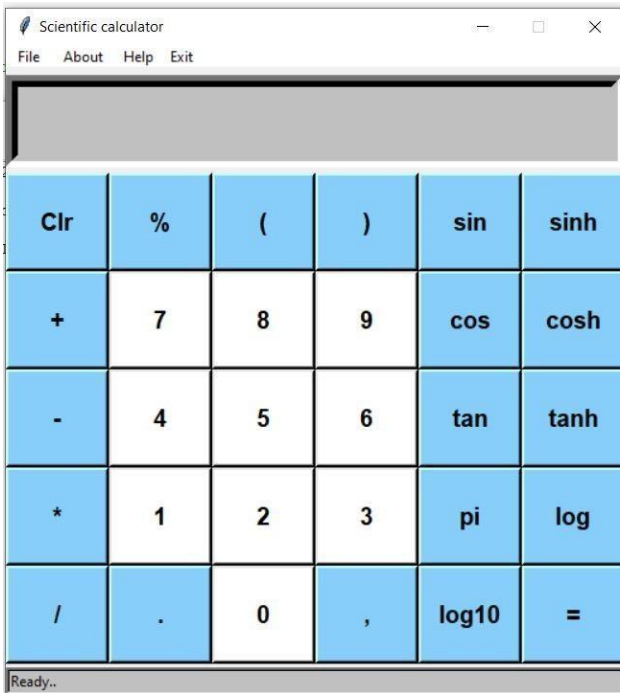
x=Button(main, option=choice,...)
Here main is the application window.

Options used:

1. text : The text to be displayed on the button.
2. font : Text font to be used for the button's label.
3. padx : Additional padding left and right of the text.
4. pady : Additional padding above and below the text.
5. borderwidth : Width of the border in pixels, default is 2.
6. fg : Normal foreground(text) color.
7. bg ; normal background color.
8. width : width of the button in letter or pixels.

Functions :

Standard :



Clr : Clears all the values entered.
+ : Addition.
- : Subtraction.
* : Multiplication.
/ : Division.
% : Modulus.
Number Pad
. : for decimals
, : separation in scientific calculations

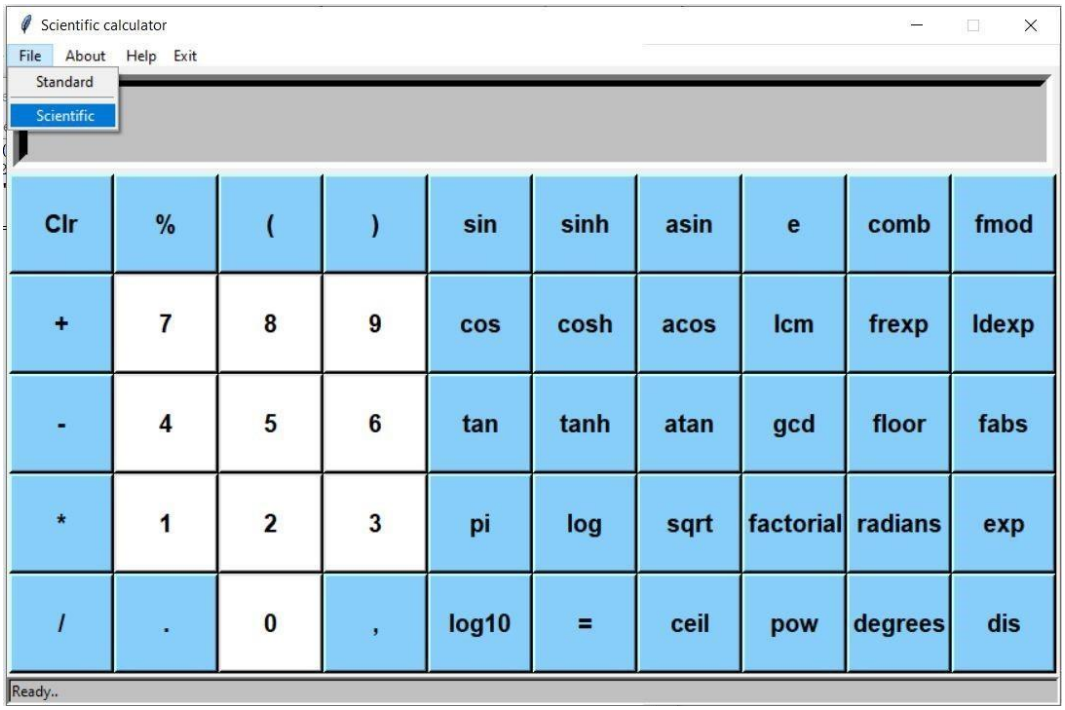
Trigonometric : Sin
, cos, tan.
Pi : 180°

Hyperbolic functions: sinh,
cosh, tanh.

Logarithmic :
Log : base e

Log10 : base 10

Scientific :



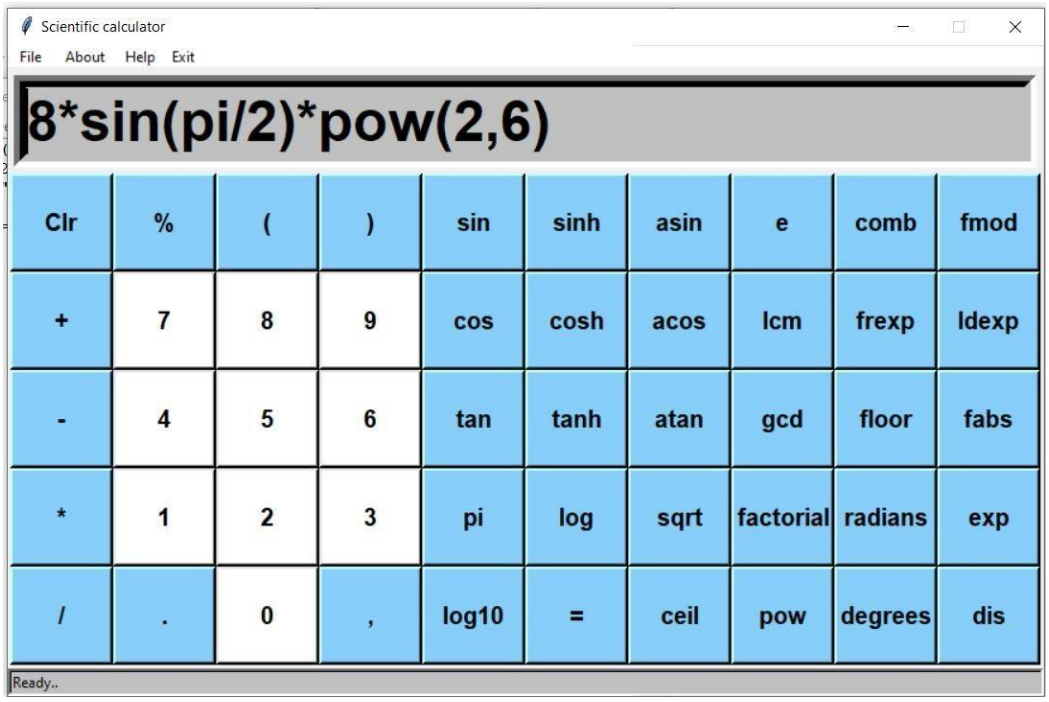
Inverse trigonometric : asin,
acos, atan.

Sqrt : square root.
Ceil : returns the smallest integer greater than or equal to x(ceil(x)) e
: 2.718281828459045
Lmc : returns least common multiple of specified integer arguments.lmc(*integers) Gcd
: returns the greatest common divisor of specified integer arguments.gcd(*integers)
Factorial : returns factorial of an integer.
Pow : pow(x,y) returns x raised to power y
Comb : comb(n,k)return the number of ways to choose k items from n items without repetition and without order
Frexp : frexp(x) returns mantissa and exponent of x as the pair(m, e).
Floor : returns the largest integer less than or equal to x

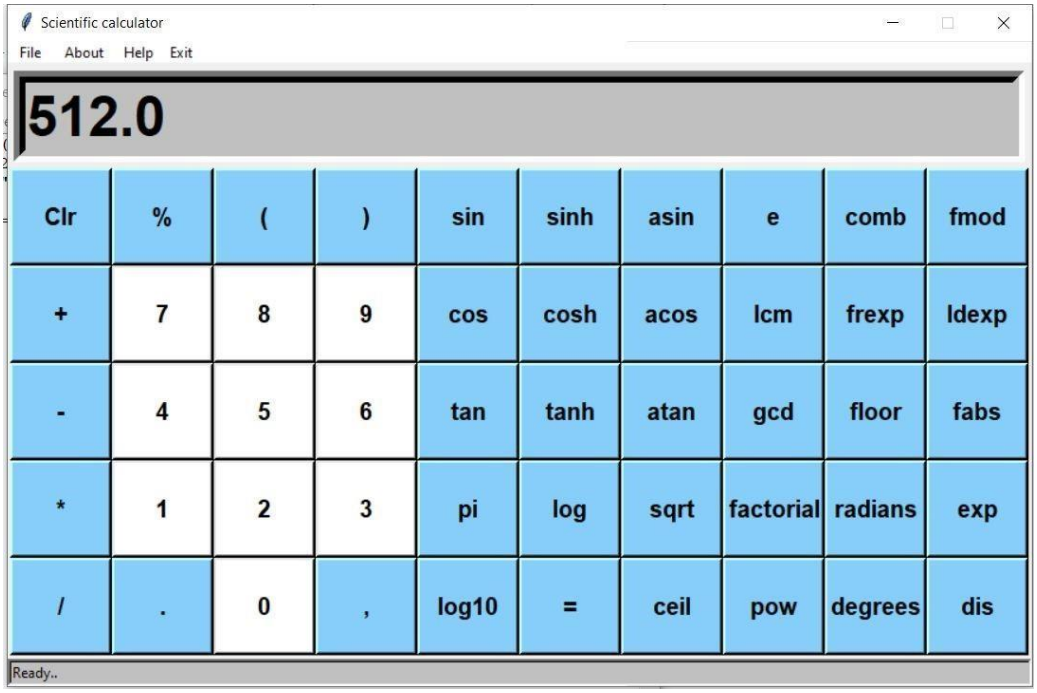
Radians : converts an angle from degree to radian
Degrees : converts an angle from radian to degrees
Fmod :fmod(x,y) is preferred while working with float values
Ldexp : inverse function of frexp()
Fabs : returns the absolute value
Exp : e raised to the power x(exp(x))
Dis : distance between 2 points

Input and output :

==========SAMPLE INPUT==========



==========SAMPLE OUTPUT==========



Conclusion :

By participating in this project we had a hands on experience on experimenting with the python modules and have increased the experience of working in a group. Using tkinter was fun and gained knowledge in it. Using the source we have we could create an application based on a scientific calculator. Experimenting with the size and colors was really fun and challenging. Overall it was a great experience and a fun way to learn and explore.