GUI Design for Scientific Calculator

**PYTHON REPORT**

Course Name: - Python Programming

Course Code: - INT 213

Submitted By:-

|  |  |  |
| --- | --- | --- |
| **Name** | **Roll No.** | **Reg No.** |
| Bhavdeep Singh | 11801591 | 7 |
| Sukhdeep Singh | 11801607 | 8 |
| Harsh Aggarwal | 11801621 | 9 |

Submitted to: - Mr. Ishan Kumar

Lovely Professional University

“School of Computer Science & Engineering”

**Introduction**

This project is replica of the scientific calculator.

A scientific calculator is designed to calculate problems in science, engineering, and mathematics. They have completely replaced slide rules in traditional applications, and are widely used in both education and professional settings.

Modern scientific calculators generally have many more features than a standard four or five-function calculator, and the feature set differs between manufacturers and models; however, the defining features of a scientific calculator include:

* Scientific notation
* Floating-point arithmetic
* Logarithmic functions, using both base 10 and [base e](https://en.wikipedia.org/wiki/Natural_logarithm)
* Trigonometric functions (some including [hyperbolic trigonometry](https://en.wikipedia.org/wiki/Hyperbolic_function))
* Exponential functions and [roots](https://en.wikipedia.org/wiki/Nth_root) beyond the [square root](https://en.wikipedia.org/wiki/Square_root)
* Quick access to constants such as [pi](https://en.wikipedia.org/wiki/Pi) and [e](https://en.wikipedia.org/wiki/E_(mathematical_constant))

Scientific calculators are used widely in situations that require quick access to certain mathematical functions, especially those that were once looked up in mathematical tables, such as trigonometric functions or logarithms. They are also used for calculations of very large or very small numbers, as in some aspects of [astronomy](https://en.wikipedia.org/wiki/Astronomy), [physics](https://en.wikipedia.org/wiki/Physics), and [chemistry](https://en.wikipedia.org/wiki/Chemistry).

They are very often required for math classes from the junior high school level through college, and are generally either permitted or required on many [standardized tests](https://en.wikipedia.org/wiki/Standardized_test) covering math and science subjects; as a result, many are sold into educational markets to cover this demand, and some high-end models include features making it easier to translate a problem on a textbook page into calculator input, e.g. by providing a method to enter an entire problem in as it is written on the page using simple formatting tools.

**Description**

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

In addition to the normal operators, it must contain the following buttons:  
  
log - base e of display  
exp - e to the display power  
sin - sine of display in radians  
cos - cosine of display in radians  
tan - tangent of display in radians  
sqrt - square root of display  
X\*\*2 - display squared

Modules/Tools

For designing this CGPA Calculator we are going to used two python libraries. The names of them are written below:

1. Tkinter

Tkinter

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

* Import the Tkinter module.
* Create the GUI application main window.
* Add one or more of the above-mentioned widgets to the GUI application.
* Enter the main event loop to take action against each event triggered by the user.

Tkinter Widgets

* Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

## Standard attributes

Let us take a look at how some of their common attributes such as sizes, colours and fonts are specified.

* [Dimensions](https://www.tutorialspoint.com/python/tk_dimensions.htm)
* [Colours](https://www.tutorialspoint.com/python/tk_colors.htm)
* [Fonts](https://www.tutorialspoint.com/python/tk_fonts.htm)
* [Anchors](https://www.tutorialspoint.com/python/tk_anchors.htm)
* [Relief styles](https://www.tutorialspoint.com/python/tk_relief.htm)
* [Bitmaps](https://www.tutorialspoint.com/python/tk_bitmaps.htm)
* [Cursors](https://www.tutorialspoint.com/python/tk_cursors.htm)

Geometry Management

All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area. Tkinter exposes the following geometry manager classes: pack, grid, and place.

* [The pack() Method](https://www.tutorialspoint.com/python/tk_pack.htm) − This geometry manager organizes widgets in blocks before placing them in the parent widget.
* [The grid() Method](https://www.tutorialspoint.com/python/tk_grid.htm) − This geometry manager organizes widgets in a table-like structure in the parent widget.
* [The place() Method](https://www.tutorialspoint.com/python/tk_place.htm) − This geometry manager organizes widgets by placing them in a specific position in the parent widget.

Screenshots of the Program

