**GUI for Scientific Calculator**

END-TERM REPORT

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

By:

|  |  |  |  |
| --- | --- | --- | --- |
| ***S.no.*** | ***Name*** | ***Roll No.*** | ***Registration no.*** |
| ***1.*** | *Ankesh Tripathi* | *16* | *11902344* |
| ***2.*** | *Ajaypal Singh* | *68* | *11909260* |
| ***3.*** | *Deepshikha Dey* | *38* | *11902067* |

**Courses Code: INT213**

**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab (India)

**Objective**

The primary objective of this project is to implement what we’ve learnt throughout our course of Python programming and use that to develop a Graphical User Interface (GUI) for Scientific calculator with all the required functionalities. This project also aims at providing a user-friendly interface to the users. This calculator can be versatile for every use as it can perform basic calculations as well as long scientific calculations. Even the person with lack of knowledge about the basic mathematics can easily use it.

**Introduction**

Since man has marched from forests to the modern lands, there has been a drastic change in the quality of life. New innovations and technologies are taking birth day by day. Science plays a vital role in making things easier. Thus this Scientific Calculator is also a part of this growing technology. This Scientific Calculator is designed to perform various calculations and is designed in such a way to meet the requirements of almost every age group of the people. This Scientific Calculator is coded in python and has a user-friendly Graphical User Interface that uses Tkinter which is a standard GUI Library in python. This project has two different modules which includes a Standard Calculator and a Scientific Calculator. Each of the module is described below.

* **Standard Calculator**

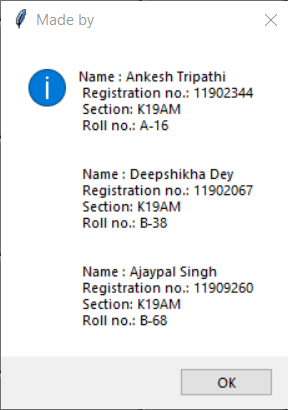
This calculator is mainly used to perform small calculations like addition, subtraction, multiplication, division, and square root. It also comes with the ‘C’ and ‘CE’ buttons. The ‘C’ button is used to clear all the entries while the ‘CE’ button is used to clear the last entry. The output always comes in floating value.

* **Scientific Calculator**

This calculator is basically used for performing scientific calculations and math functions like logarithm, sin, cosine, tan, exponential, pie, modulus, degree, power. There is a dedicated button for finding log with base as 10 or 2. Also pie has two separate buttons **π** and 2**π**. It also shows the output in floating points.

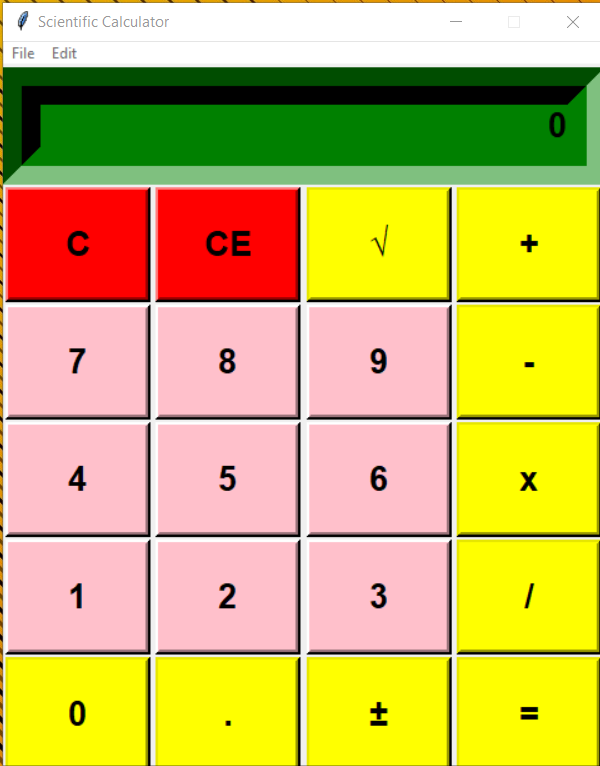
* **Made By Button**

This Made by button is present in the welcome page under the file dropdown. It is given to provide the details of each of the members in the group. On pressing this button showinfo message box will be called and it will display the name, registration number, roll number and section of each of the member.

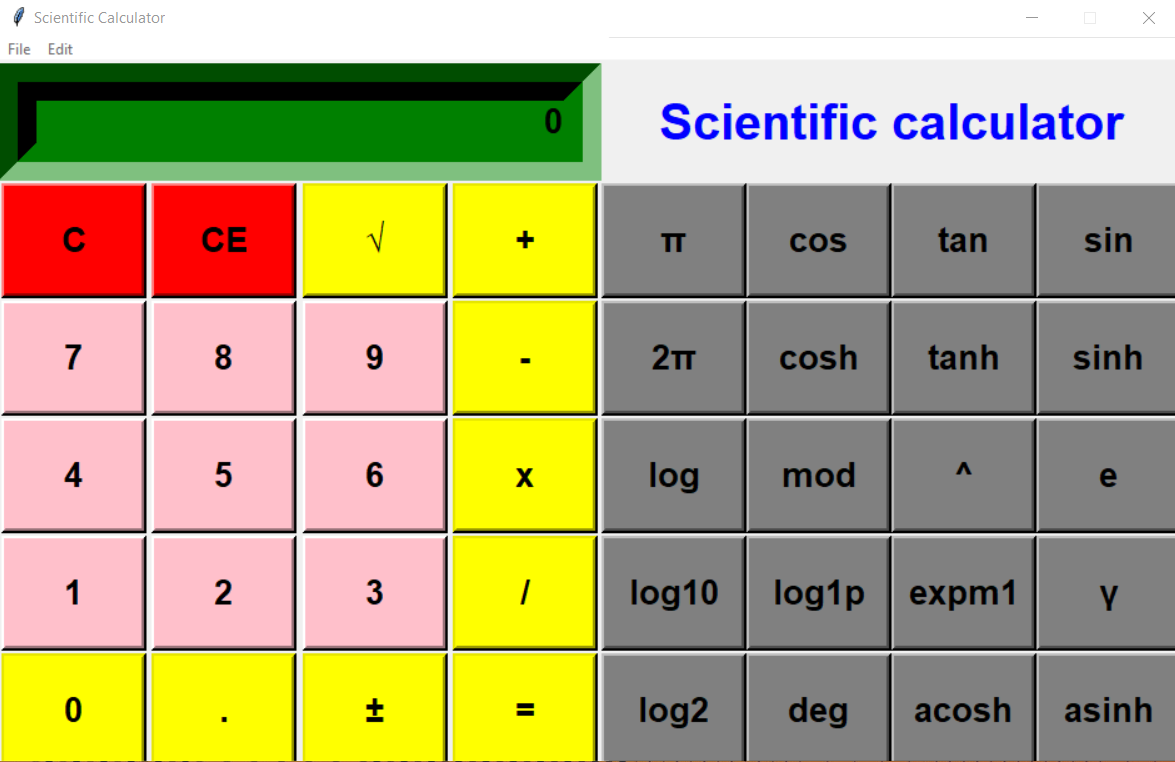


**GUI Screenshots:**

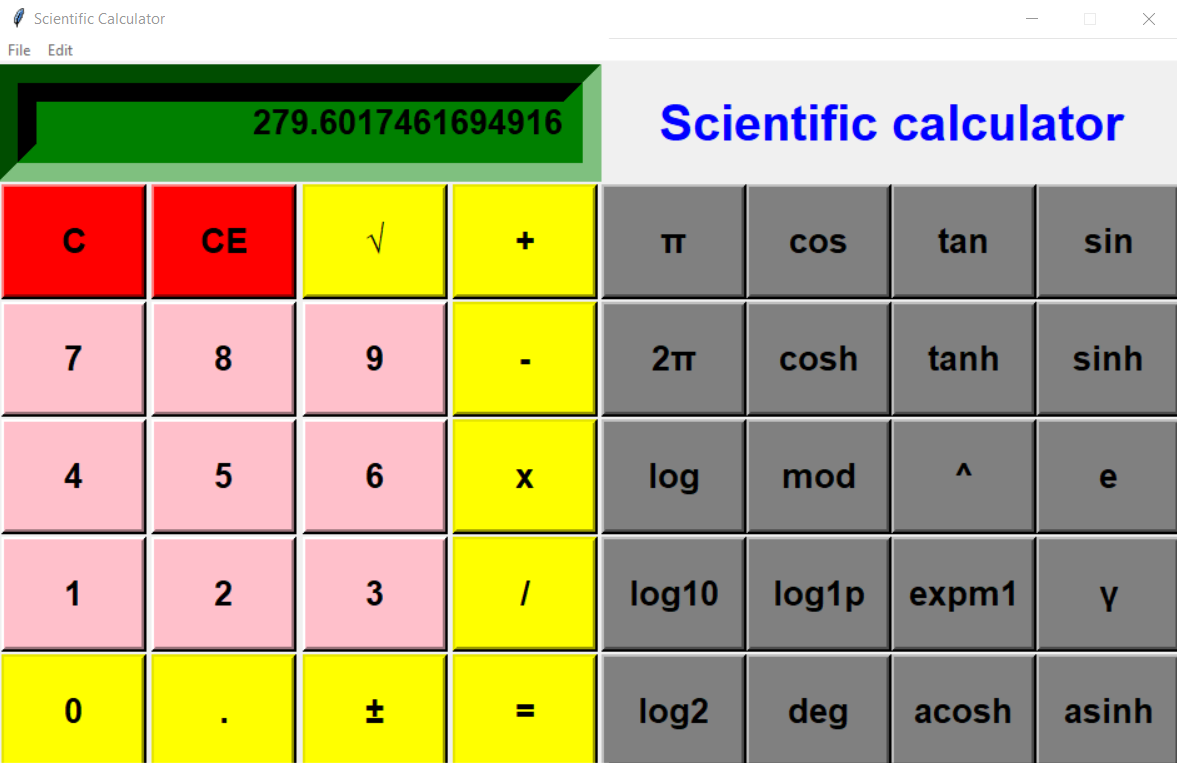
**1.STANDARD CALCULATOR:**



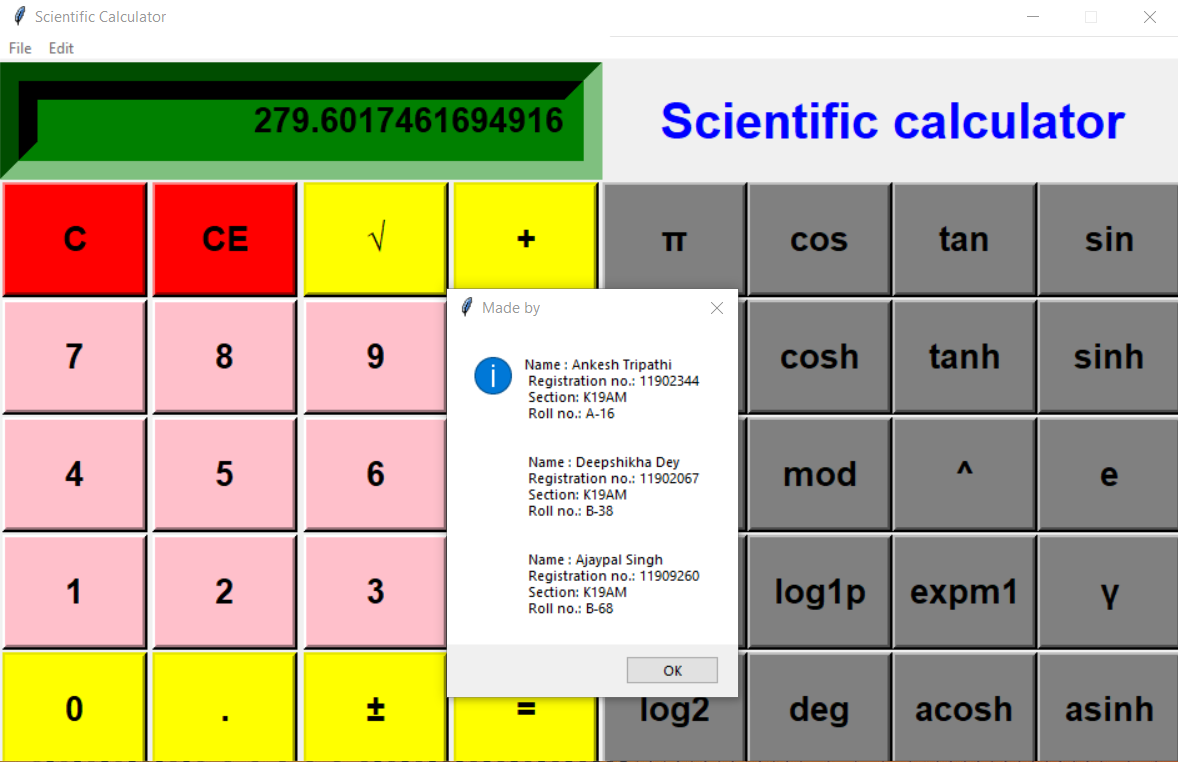
**2.SCIENTIFIC CALCULATOR:**



**3.SAMPLE CALCULATIONS:**



**4.MESSAGE BOX:**



**5.Exit:**



**Source Code**

from tkinter import \*

import math

import parser

#primary purpose for this interface is to allow Python code to edit the parse tree of a Python expression and create executable code from this

import tkinter.messagebox

#MessageBox module is used to display message boxes in the output

root = Tk()

root.title("Scientific Calculator")

root.configure(bg = 'blue')

root.resizable(width=False, height=False) # prohibit resizing of the window or it is used to create a fixed side window

root.geometry("480x560")

calc = Frame(root)

calc.grid() #organizes widgets in a table-like structure in the parent widget

#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++#

class Calc():

def \_\_init\_\_(self):

self.total=0

self.current=' '

self.input\_value=True

self.check\_sum=False

self.op=''

self.result=False

def numenter(self,num): #this function is to take user input in the calculator

self.result=False

firstnum=txtdisplay.get()

secondnum=str(num)

if self.input\_value:

self.current = secondnum

self.input\_value=False

else:

if secondnum == '.':

if secondnum in firstnum:

return

self.current = firstnum + secondnum

self.display(self.current)

def sum\_of\_total(self):

self.result=True

self.current=float(self.current)

if self.check\_sum==True:

self.valid\_function()

else:

self.total= float(txtdisplay.get())

def display(self, value):

txtdisplay.delete(0, END)

txtdisplay.insert(0, value)

def valid\_function(self): # for the standard operations

if self.op == "add":

self.total += self.current

if self.op == "sub":

self.total -= self.current

if self.op == "mul":

self.total \*= self.current

if self.op == "div":

self.total /= self.current

if self.op == "mod":

self.total %= self.current

self.input\_value = True

self.check\_sum = False

self.display(self.total)

def operation(self,op):

self.current = float(self.current)

if self.check\_sum:

self.valid\_function()

elif not self.result:

self.total=self.current

self.input\_value = True

self.check\_sum = True

self.op = op

self.result = False

def Clear\_entry(self): # function for clear the current display

self.result = False

self.current = "0"

self.display(0)

self.input\_value = True

def All\_clear(self): # function to clear all the data

self.Clear\_entry()

self.total = 0

def pi(self): # fuction for pi

self.result=False

self.current=math.pi

self.display(self.current)

def tau(self): # fuction for 2 pi

self.result=False

self.current = math.tau

self.display(self.current)

def e(self): # function for e

self.result=False

self.current=math.e

self.display(self.current)

def mathPM(self): # function for plus minus

self.result = False

self.current = -(float(txtdisplay.get()))

self.display(self.current)

def square(self): # function for square of any no.

self.result = False

self.current = math.sqrt(float(txtdisplay.get()))

self.display(self.current)

def cos(self): # function for cos

self.result = False

self.current = math.cos(math.radians(float(txtdisplay.get())))

self.display(self.current)

def cosh(self): # function for hyperbolic cos i.e cosh

self.result = False

self.current = math.cosh(math.radians(float(txtdisplay.get())))

self.display(self.current)

def tan(self): # function for tan

self.result = False

self.current = math.tan(math.radians(float(txtdisplay.get())))

self.display(self.current)

def tanh(self): # function for hyperbolic tan i.e tanh

self.result = False

self.current = math.tanh(math.radians(float(txtdisplay.get())))

self.display(self.current)

def sin(self): # function for sin

self.result = False

self.current = math.sin(math.radians(float(txtdisplay.get())))

self.display(self.current)

def sinh(self): # function for hyperbolic sin i.e sinh

self.result = False

self.current = math.sinh(math.radians(float(txtdisplay.get())))

self.display(self.current)

def log(self): # function for log

self.result = False

self.current = math.log(float(txtdisplay.get()))

self.display(self.current)

def exp(self): # function for exponent

self.result = False

self.current = math.exp(float(txtdisplay.get()))

self.display(self.current)

def acosh(self): # function for acosh

self.result = False

self.current = math.acosh(float(txtdisplay.get()))

self.display(self.current)

def asinh(self): # function for asinh

self.result = False

self.current = math.asinh(float(txtdisplay.get()))

self.display(self.current)

def expm1(self): # function for exponential plus minus

self.result = False

self.current = math.expm1(float(txtdisplay.get()))

self.display(self.current)

def lgamma(self): # function for gamma

self.result = False

self.current = math.lgamma(float(txtdisplay.get()))

self.display(self.current)

def degrees(self): # function for degree

self.result = False

self.current = math.degrees(float(txtdisplay.get()))

self.display(self.current)

def log2(self): # function for log2 (log base 2)

self.result = False

self.current = math.log2(float(txtdisplay.get()))

self.display(self.current)

def log10(self): # function for log10 (log base 10)

self.result = False

self.current = math.log10(float(txtdisplay.get()))

self.display(self.current)

def log1p(self): # function for log1p (log base 1p)

self.result = False

self.current = math.log1p(float(txtdisplay.get()))

self.display(self.current)

added\_value = Calc()

#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++#

txtdisplay = Entry(calc, font=('Helvetica',20,'bold'),bg= 'green', bd=30, width= 28,justify=RIGHT)

txtdisplay.grid(row=0, column=0, columnspan=4, pady=1)

txtdisplay.insert(0,"0")

numberpad = "789456123"

i=0

btn = []

for j in range(2,5):

for k in range(3):

btn.append(Button(calc, width = 6, height = 2, bg="pink", font=('Helvetica',20,'bold'), bd=4, text = numberpad[i]))

btn[i].grid(row=j, column = k, pady =1)

btn[i]["command"] = lambda x=numberpad[i]:added\_value.numenter(x)

i+=1

#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++#

#buttons for Standard part

btnClear = Button(calc, text=chr(67),width = 6, height = 2, bg="red", font=('Helvetica',20,'bold'),

bd=4, command=added\_value.Clear\_entry).grid(row=1, column = 0, pady =1)

btnAllClear = Button(calc, text=chr(67)+chr(69),width = 6, height = 2, bg="red", font=('Helvetica',20,'bold'),

bd=4, command=added\_value.All\_clear).grid(row=1, column = 1, pady =1)

btnsq = Button(calc, text="\u221A",width = 6, height = 2, bg="yellow", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.square).grid(row=1, column = 2, pady =1)

btnAdd = Button(calc, text="+",width = 6, height = 2, bg="yellow", font=('Helvetica',20,'bold'),

bd=4,command = lambda:added\_value.operation("add")).grid(row=1, column = 3, pady =1)

btnSubtract = Button(calc, text="-",width = 6, height = 2, bg="yellow", font=('Helvetica',20,'bold'),

bd=4,command = lambda:added\_value.operation("sub")).grid(row=2, column = 3, pady =1)

btnMul = Button(calc, text="x",width = 6, height = 2, bg="yellow", font=('Helvetica',20,'bold'),

bd=4,command = lambda:added\_value.operation("mul")).grid(row=3, column = 3, pady =1)

btnDiv = Button(calc, text="/",width = 6, height = 2, bg="yellow", font=('Helvetica',20,'bold'),

bd=4,command = lambda:added\_value.operation("div")).grid(row=4, column = 3, pady =1)

btnzero = Button(calc, text="0",width = 6, height = 2, bg="yellow", font=('Helvetica',20,'bold'),

bd=4,command = lambda:added\_value.numenter(0)).grid(row=5, column = 0, pady =1)

btndot = Button(calc, text=".",width = 6, height = 2, bg="yellow", font=('Helvetica',20,'bold'),

bd=4,command = lambda:added\_value.numenter(".")).grid(row=5, column = 1, pady =1)

btnPM = Button(calc,text = chr(177),width = 6, height = 2, bg="yellow", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.mathPM).grid(row=5, column = 2, pady =1)

btnEqual = Button(calc, text="=",width = 6, height = 2, bg="yellow", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.sum\_of\_total).grid(row=5, column = 3, pady =1)

#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++#

#buttons for scientific part

btnPi = Button(calc, text="π",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.pi).grid(row=1, column = 4, pady =1)

btnCos = Button(calc, text="cos",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.cos).grid(row=1, column = 5, pady =1)

btnTan = Button(calc,text = "tan",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.tan).grid(row=1, column = 6, pady =1)

btnSin = Button(calc, text="sin",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.sin).grid(row=1, column = 7, pady =1)

btn2Pi = Button(calc, text="2π",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.tau).grid(row=2, column = 4, pady =1)

btnCosh = Button(calc, text="cosh",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.cosh).grid(row=2, column = 5, pady =1)

btnTanh = Button(calc,text = "tanh",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.tanh).grid(row=2, column = 6, pady =1)

btnSinh = Button(calc, text="sinh",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.sinh).grid(row=2, column = 7, pady =1)

btnlog = Button(calc, text="log",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.log).grid(row=3, column = 4, pady =1)

btnmod = Button(calc, text="mod",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command = lambda:added\_value.operation("mod")).grid(row=3, column = 5, pady =1)

btnExp = Button(calc,text = "^",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.exp).grid(row=3, column = 6, pady =1)

btnE = Button(calc, text="e",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.e).grid(row=3, column = 7, pady =1)

btnlog10 = Button(calc, text="log10",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.log10).grid(row=4, column = 4, pady =1)

btnlog1p = Button(calc, text="log1p",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.log1p).grid(row=4, column = 5, pady =1)

btnExpPM = Button(calc,text = "expm1",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.expm1).grid(row=4, column = 6, pady =1)

btngamma = Button(calc, text="γ",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.lgamma).grid(row=4, column = 7, pady =1)

btnlog2 = Button(calc, text="log2",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.log2).grid(row=5, column = 4, pady =1)

btndeg = Button(calc, text="deg",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.degrees).grid(row=5, column = 5, pady =1)

btncosh = Button(calc,text = "acosh",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.acosh).grid(row=5, column = 6, pady =1)

btnsinh = Button(calc, text="asinh",width = 6, height = 2, bg="grey", font=('Helvetica',20,'bold'),

bd=4,command=added\_value.asinh).grid(row=5, column = 7, pady =1)

lblDisplay = Label(calc, text = "Scientific calculator", font = ('Helvetica', 30,'bold'),fg='blue',justify=CENTER)

lblDisplay.grid(row=0, column=4, columnspan=4)

#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++#

# functions for sub menus

def iexit():

iexit = tkinter.messagebox.askyesno("Scientific Calculator","Do you want to Exit?")

if iexit>0:

root.destroy()

# destroy() method in Tkinter destroys a widget.

# Also when a process is complete by some user action we need to destroy the GUI components to free the memory as well as clear the screen

return

def scientific():

root.resizable(width=False,height=False) # prohibit resizing of the window or it is used to create a fixed side window

root.geometry("944x560")

def standard():

root.resizable(width=False,height=False) # prohibit resizing of the window or it is used to create a fixed side window

root.geometry("480x560")

def madeby():

madeby = tkinter.messagebox.showinfo("Made by","Name : Ankesh Tripathi\n Registration no.: 11902344\n "

"Section: K19AM\n Roll no.: A-16\n\n\n "

"Name : Deepshikha Dey\n Registration no.: 11902067\n "

"Section: K19AM\n Roll no.: B-38\n\n\n "

"Name : Ajaypal Singh\n Registration no.: 11909260\n "

"Section: K19AM\n Roll no.: B-68")

menubar=Menu(calc) # creating menu

filemenu = Menu(menubar,tearoff = 0) # tearoff allows you to detach menus for the main window creating floating menus

menubar.add\_cascade(label = 'File', menu = filemenu) # add\_cascade allows us to add create kind of menus that can be used for our application

filemenu.add\_command(label="Standard", command = standard) # add\_command is used for the tasks with are there in menu

filemenu.add\_command(label="Scientific", command = scientific)

filemenu.add\_separator() # add\_seperator is used to create a line in the menu

filemenu.add\_command(label = "Made by", command = madeby)

filemenu.add\_separator()

filemenu.add\_command(label="Exit",command = iexit)

editmenu = Menu(menubar,tearoff = 0) # tearoff allows you to detach menus for the main window creating floating menus

menubar.add\_cascade(label = 'Edit', menu = editmenu) # add\_cascade allows us to add create kind of menus that can be used for our application

editmenu.add\_command(label="Cut") # add\_command is used for the tasks with are there in menu

editmenu.add\_command(label="Copy")

editmenu.add\_separator() # add\_seperator is used to create a line in the menu

editmenu.add\_command(label="Paste")

root.config(menu=menubar)

root.mainloop()

**Results:**

We finally got the end product as a ‘Scientific Calculator’ which includes a standard as well as a scientific calculator. We learnt how to make a GUI using Tkinter in python. It also helped us in learning various GUI widgets like the menu buttons, message boxes, list boxes, buttons etc.

This calculator offers a number of tools which help you in solving various problems from simple addition/subtraction to long and lengthy exponential calculations. The following are the results which we yielded from our project.

* Standard Calculator performs the basic calculations like addition, subtraction, multiplication, division, square root etc. It is for the general-purpose use.
* Scientific Calculator performs the calculations in science, engineering, and mathematics. It is used both in educational and professional settings.

**References:**

* <https://www.w3schools.com/python/>
* [https://docs.python.org/3/library/functions.html](https://www.python-course.eu/index.php)
* <https://www.python-course.eu/index.php>
* <https://www.geeksforgeeks.org/sql-using-python/>
* <https://python-forum.io/index.php>
* <https://stackoverflow.com/>
* [www.tutorialspoint.com](http://www.tutorialspoint.com/)
* [www.reddit.com](http://www.reddit.com/)