2) wxPython for GUI application. As part of following the video you installed wxPython. I am providing you with a sample.py application that runs and creates a template for you to develop you own application with the following requirements. Here is a link to the [wxpython documentation (Links to an external site.)](https://docs.wxpython.org/" \t "_blank) as a reference.

The application is to allow you to select a text file that contain any number of sentences, once the file is selected, your application is to count the frequency of the words found in the document and provide a popup that show the top three most detected words and their respective count. Please submit a screenshot of the application producing the results and a copy of your wxpython program.

[Sample python codePreview the document](https://fiu.instructure.com/courses/64901/files/8251454/download?wrap=1)

Ans Code:

import wx

from tkinter.filedialog import askopenfile

from tkinter import Tk

import tkinter.messagebox as tkmb

from collections import Counter

class windowClass(wx.Frame):

def \_\_init\_\_(self, \*args, \*\*kwargs):

super(windowClass, self).\_\_init\_\_(\*args, \*\*kwargs)

self.Centre()

self.basicGUI()

def basicGUI(self):

panel = wx.Panel(self)

menuBar = wx.MenuBar()

fileButton = wx.Menu()

editButton = wx.Menu()

importItem = wx.Menu()

viewItem = wx.Menu()

openItem = fileButton.Append(wx.ID\_ANY, 'Open File', 'Select a file ...')

exitItem = fileButton.Append(wx.ID\_EXIT, 'Exit', 'Exit sample ...')

menuBar.Append(fileButton, 'File')

self.SetMenuBar(menuBar)

self.Bind(wx.EVT\_MENU, self.OpenMode, openItem)

self.Bind(wx.EVT\_MENU, self.Quit, exitItem)

self.statusbar = self.CreateStatusBar(1)

self.statusbar.SetStatusText('Waiting To select Open')

self.SetTitle('Open a File')

self.Show(True)

def Quit(self, e):

yesNoBox = wx.MessageDialog(None, 'Are you sure you want to Quit?', 'Question', wx.YES\_NO)

yesNoAnswer = yesNoBox.ShowModal()

yesNoBox.Destroy()

if yesNoAnswer == wx.ID\_YES:

self.Close()

def OpenMode(self,e):

Tk().withdraw()

file = askopenfile(mode ='r', filetypes =[('Text Files', '\*.txt')])

if file is not None:

count = Counter(word for line in file

for word in line.split())

words =(count.most\_common(3))

tkmb.showinfo("Common words", words)

def main():

app = wx.App()

windowClass(None, 0, size=(500,400))

app.MainLoop()

main()

Final Product:

