# Python program to create a GUI mark sheet using tkinter

```
import tkinter as tk

master = tk.Tk()

master.title("MARKSHEET")

master.geometry("700x250")


e1 = tk.Entry(master)

e2 = tk.Entry(master)

e3 = tk.Entry(master)

e4 = tk.Entry(master)

e5 = tk.Entry(master)

e6 = tk.Entry(master)

e7 = tk.Entry(master)


def display():

        tot = 0

        if e4.get() == "A":

                tk.Label(master, text="40").grid(row=3, column=4)

                tot += 40

        if e4.get() == "B":

                tk.Label(master, text="36").grid(row=3, column=4)

                tot += 36

        if e4.get() == "C":

                tk.Label(master, text="32").grid(row=3, column=4)

                tot += 32



        if e4.get() == "D":

                tk.Label(master, text="28").grid(row=3, column=4)

                tot += 28
        if e4.get() == "P":
                tk.Label(master, text="24").grid(row=3, column=4)
                tot += 24


        if e4.get() == "F":
                tk.Label(master, text="0").grid(row=3, column=4)
                tot += 0
        if e5.get() == "A":
                tk.Label(master, text="40").grid(row=4, column=4)
                tot += 40
        if e5.get() == "B":
                tk.Label(master, text="36").grid(row=4, column=4)
                tot += 36
        if e5.get() == "C":
                tk.Label(master, text="32").grid(row=4, column=4)
                tot += 32
        if e5.get() == "D":
                tk.Label(master, text="28").grid(row=4, column=4)
                tot += 28
        if e5.get() == "P":
                tk.Label(master, text="28").grid(row=4, column=4)
                tot += 24
        if e5.get() == "F":
                tk.Label(master, text="0").grid(row=4, column=4)
                tot += 0


        if e6.get() == "A":
                tk.Label(master, text="30").grid(row=5, column=4)
                tot += 30
        if e6.get() == "B":
                tk.Label(master, text="27").grid(row=5, column=4)
```

```
            tot += 27
    if e6.get() == "C":
            tk.Label(master, text="24").grid(row=5, column=4)
            tot += 24
    if e6.get() == "D":
            tk.Label(master, text="21").grid(row=5, column=4)
            tot += 21
    if e6.get() == "P":
            tk.Label(master, text="28").grid(row=5, column=4)
            tot += 24
    if e6.get() == "F":
            tk.Label(master, text="0").grid(row=5, column=4)
            tot += 0

    if e7.get() == "A":
            tk.Label(master, text="40").grid(row=6, column=4)
            tot += 40
    if e7.get() == "B":
            tk.Label(master, text="36").grid(row=6, column=4)
            tot += 36
    if e7.get() == "C":
            tk.Label(master, text="32").grid(row=6, column=4)
            tot += 32
    if e7.get() == "D":
            tk.Label(master, text="28").grid(row=6, column=4)
            tot += 28
    if e7.get() == "P":
            tk.Label(master, text="28").grid(row=6, column=4)
            tot += 24
    if e7.get() == "F":
            tk.Label(master, text="0").grid(row=6, column=4)
```

```
                    tot += 0

          # to display total credits
          tk.Label(master, text=str(tot)).grid(row=7, column=4)

          # to display SGPA
          tk.Label(master, text=str(tot/15)).grid(row=8, column=4)

# end of display function

# label to enter name
tk.Label(master, text="Name").grid(row=0, column=0)

# label for registration number
tk.Label(master, text="Reg.No").grid(row=0, column=3)

# label for roll Number
tk.Label(master, text="Roll.No").grid(row=1, column=0)

# labels for serial numbers
tk.Label(master, text="Srl.No").grid(row=2, column=0)
tk.Label(master, text="1").grid(row=3, column=0)
tk.Label(master, text="2").grid(row=4, column=0)
tk.Label(master, text="3").grid(row=5, column=0)
tk.Label(master, text="4").grid(row=6, column=0)

# labels for subject codes
tk.Label(master, text="Subject").grid(row=2, column=1)
tk.Label(master, text="CS 201").grid(row=3, column=1)
tk.Label(master, text="CS 202").grid(row=4, column=1)
tk.Label(master, text="MA 201").grid(row=5, column=1)
tk.Label(master, text="EC 201").grid(row=6, column=1)


# label for grades
tk.Label(master, text="Grade").grid(row=2, column=2)
e4.grid(row=3, column=2)
e5.grid(row=4, column=2)
e6.grid(row=5, column=2)
e7.grid(row=6, column=2)
# labels for subject credits
tk.Label(master, text="Sub Credit").grid(row=2, column=3)
tk.Label(master, text="4").grid(row=3, column=3)
tk.Label(master, text="4").grid(row=4, column=3)
tk.Label(master, text="3").grid(row=5, column=3)
tk.Label(master, text="4").grid(row=6, column=3)
tk.Label(master, text="Credit obtained").grid(row=2, column=4)
# taking entries of name, reg, roll number respectively
e1 = tk.Entry(master)
e2 = tk.Entry(master)
e3 = tk.Entry(master)
# organizing them in the grid
e1.grid(row=0, column=1)
e2.grid(row=0, column=4)
e3.grid(row=1, column=1)
# button to display all the calculated credit scores and sgpa
button1 = tk.Button(master, text="submit", bg="green", command=display)
button1.grid(row=8, column=1)


tk.Label(master, text="Total credit").grid(row=7, column=3)
tk.Label(master, text="SGPA").grid(row=8, column=3)
master.mainloop()
```

GUI to Shutdown, Restart and Logout from the PC using Python

```
# import modules
from tkinter import *
import os

# user define function
def shutdown():
	return os.system("shutdown /s /t 1")

def restart():
	return os.system("shutdown /r /t 1")

def logout():
	return os.system("shutdown -l")


# tkinter object
master = Tk()

# background set to grey
master.configure(bg='light grey')

Button(master, text="Shutdown", command=shutdown).grid(row=0)
Button(master, text="Restart", command=restart).grid(row=1)
Button(master, text="Log out", command=logout).grid(row=2)

mainloop()
```

Build an Application to Search Installed Application using Python

```
# import modules

from tkinter import *

import winapps

# function to attach output

def app():


        for item in winapps.search_installed(e.get()):

                name.set(item.name)

                version.set(item.version)

                Install_date.set(item.install_date)

                publisher.set(item.publisher)

                uninstall_string.set(item.uninstall_string)




# object of tkinter

# and background set for grey

master = Tk()

master.configure(bg='light grey')


# Variable Classes in tkinter

name = StringVar()

version = StringVar()

Install_date = StringVar()

publisher = StringVar()

uninstall_string = StringVar()


Label(master, text="Enter App name : ",

        bg="light grey").grid(row=0, sticky=W)
```

```
Label(master, text="Name : ",
            bg="light grey").grid(row=2, sticky=W)
Label(master, text="Version :",
            bg="light grey").grid(row=3, sticky=W)
Label(master, text="Install date :",
            bg="light grey").grid(row=4, sticky=W)
Label(master, text="publisher :",
            bg="light grey").grid(row=5, sticky=W)
Label(master, text="Uninstall string :",
            bg="light grey").grid(row=6, sticky=W)

# Creating label for class variable
# name using widget Entry
Label(master, text="", textvariable=name,
            bg="light grey").grid(row=2, column=1, sticky=W)
Label(master, text="", textvariable=version,
            bg="light grey").grid(row=3, column=1, sticky=W)
Label(master, text="", textvariable=Install_date,
            bg="light grey").grid(row=4, column=1, sticky=W)
Label(master, text="", textvariable=publisher,
            bg="light grey").grid(row=5, column=1, sticky=W)
Label(master, text="", textvariable=uninstall_string,
            bg="light grey").grid(row=6, column=1, sticky=W)

e = Entry(master, width=30)
e.grid(row=0, column=1)
# creating a button using the widget
b = Button(master, text="Show", command=app, bg="Blue")
b.grid(row=0, column=2, columnspan=2, rowspan=2, padx=5, pady=5,)
mainloop()
```

Create a Simple Two Player Game using Turtle in Python

```
import random

import turtle


# function to check whether turtle

# is in Screen or not

def isInScreen(win, turt):


	# getting the end points of turtle screen

	leftBound = -win.window_width() / 2

	rightBound = win.window_width() / 2

	topBound = win.window_height() / 2

	bottomBound = -win.window_height() / 2


	# getting the current position of the turtle

	turtleX = turt.xcor()

	turtleY = turt.ycor()


	# variable to store whether in screen or not

	stillIn = True


	# condition to check whether in screen or not

	if turtleX > rightBound or turtleX < leftBound:

		stillIn = False

	if turtleY > topBound or turtleY < bottomBound:

		stillIn = False


	# returning the result

	return stillIn
```

```
# function to check whether both turtle have
# different position or not
def sameposition(Red, Blue):
        if Red.pos() == Blue.pos():
                return False
        else:
                return True

# main function
def main():

        # screen initialization for turtle
        wn = turtle.Screen()

        # Turtle Red initialization
        # instantiate a new turtle object
        # called 'Red'
        Red = turtle.Turtle()

        # set pencolor as red
        Red.pencolor("red")

        # set pensize as 5
        Red.pensize(5)

        # set turtleshape as turtle
        Red.shape('turtle')
        pos = Red.pos()

        # Turtle Blue initialization
        # instantiate a new turtle object
        # called 'Blue'
        Blue = turtle.Turtle()

        # set pencolor as blue
        Blue.pencolor("blue")

        # set pensize as 5
        Blue.pensize(5)

        # set turtleshape as turtle
        Blue.shape('turtle')

        # make the turtle invisible
        Blue.hideturtle()

        # don't draw when turtle moves
        Blue.penup()

        # move the turtle to a location 50
        # units away from Red
        Blue.goto(pos[0]+50, pos[1])

        # make the turtle visible
        Blue.showturtle()

        # draw when the turtle moves
        Blue.pendown()

        # variable to store whether turtles
        # are in screen or not
        mT = True
```

```
jT = True

# loop for the game
while mT and jT and sameposition(Red, Blue):

    # coin flip for Red
    coinRed = random.randrange(0, 2)

    # angle for Red
    # random.randrange(0, 180)
    angleRed = 90

    # condition for left or right
    # based on coin
    if coinRed == 0:
        Red.left(angleRed)
    else:
        Red.right(angleRed)

    # coin flip for Blue
    coinBlue = random.randrange(0, 2)

    # angle for Blue
    # random.randrange(0, 180)
    angleBlue = 90

    # condition for left or right based
    # on coin
    if coinBlue == 0:
        Blue.left(angleBlue)
    else:
```

```
                Blue.right(angleBlue)

        # draw for Red
        Red.forward(50)

        # draw for Blue
        Blue.forward(50)

        # checking whether turtles are in the
        # screen or not
        mT = isInScreen(wn, Blue)
        jT = isInScreen(wn, Red)

# set pencolor for Blue and Red as black
Red.pencolor("black")
Blue.pencolor("black")

# condition check for draw or win
if jT == True and mT == False:
        # writing results
        Red.write("Red Won", True, align="center",
                        font=("arial", 15, "bold"))

elif mT == True and jT == False:

        # writing results
        Blue.write("Blue Won", True, align="center",
                        font=("arial", 15, "bold"))
else:
        # writing results
        Red.write("Draw", True, align="center",
                                        font=("arial", 15, "bold"))

                    Blue.write("Draw", True, align="center",

                                        font=("arial", 15, "bold"))


          wn.exitonclick()

main()
```