**Eg.1**

from tkinter import \*

import tkinter.messagebox

def validate():

name= entry\_1.get()

email= entry\_2.get()

country= c.get()

programming= var2.get()

gender= var.get()

if (name=="" or email=="" or country== 'Select your country' or gender == 0):

tkinter.messagebox.showinfo('Invalid Message Alert',"Fields cannot be left empty!")

else:

tkinter.messagebox.showinfo('Success Message',"Successfully registered!")

root = Tk()

root.geometry('500x500')

root.title("Registration Form")

label\_0 = Label(root, text="Registration form",width=20,font=("bold", 20))

label\_0.place(x=90,y=53)

label\_1 = Label(root, text="Full Name",width=20,font=("bold", 10))

label\_1.place(x=80,y=130)

entry\_1 = Entry(root)

entry\_1.place(x=240,y=130)

label\_2 = Label(root, text="Email",width=20,font=("bold", 10))

label\_2.place(x=68,y=180)

entry\_2 = Entry(root)

entry\_2.place(x=240,y=180)

label\_3 = Label(root, text="Gender",width=20,font=("bold", 10))

label\_3.place(x=70,y=230)

var = IntVar()

Radiobutton(root, text="Male",padx = 5, variable=var, value=1).place(x=235,y=230)

Radiobutton(root, text="Female",padx = 20, variable=var, value=2).place(x=290,y=230)

label\_4 = Label(root, text="Country",width=20,font=("bold", 10))

label\_4.place(x=70,y=280)

list1 = ['Myanmar','Canada','India','UK','Nepal','Iceland','South Africa'];

c=StringVar()

droplist=OptionMenu(root,c, \*list1)

droplist.config(width=15)

c.set('Select your country')

droplist.place(x=240,y=280)

label\_4 = Label(root, text="Programming",width=20,font=("bold", 10))

label\_4.place(x=85,y=330)

var1 = IntVar()

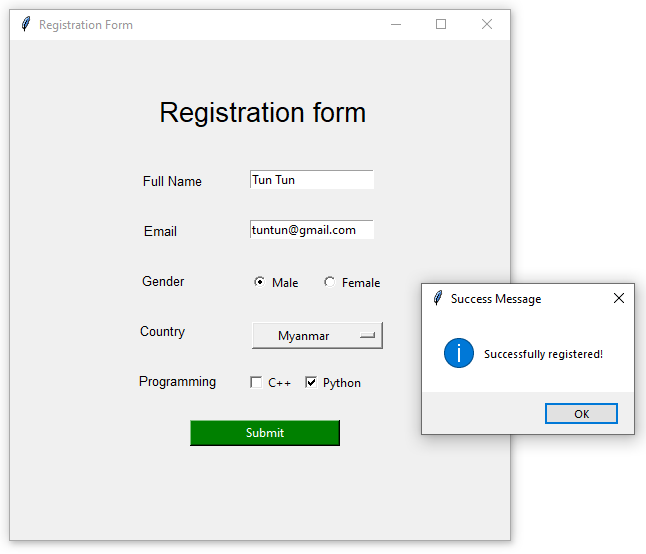
Checkbutton(root, text="C++", variable=var1).place(x=235,y=330)

var2 = IntVar()

Checkbutton(root, text="Python", variable=var2).place(x=290,y=330)

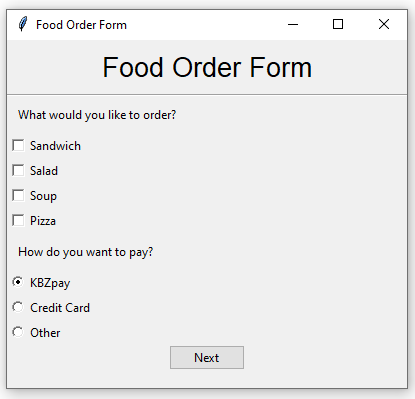
Button(root, text='Submit',width=20,bg='green',fg='white', command = validate).place(x=180,y=380)

root.mainloop()



**Eg. 2**

When the next\_button is clicked, the printResults() method is called. This prints the text from selected checkbuttons and radiobutton and closes the program.



from tkinter import (Tk, ttk, Label, Frame, Button, Checkbutton,

Radiobutton, IntVar, HORIZONTAL)

class SimpleGUI(Tk):

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

super().\_\_init\_\_()

self.initializeUI()

def initializeUI(self):

self.title("Food Order Form")

self.minsize(300, 200) # width, height

self.geometry("400x350+50+50")

self.setupWindow()

def setupWindow(self):

""" Set up the widgets."""

title = Label(self, text="Food Order Form",

font=('Helvetica', 20), bd=10)

title.pack()

line = ttk.Separator(self, orient=HORIZONTAL)

line.pack(fill='x')

order\_label = Label(self, text="What would you like to order?", bd=10)

order\_label.pack(anchor='w')

foods\_list = ["Sandwich", "Salad", "Soup", "Pizza"]

self.foods\_dict = {}

# Populate the dictionary with checkbutton widgets

for i, food\_item in enumerate(foods\_list):

# Set the text for each checkbutton

self.foods\_dict[food\_item] = Checkbutton(self, text=food\_item)

# Create a new instance of IntVar() for each checkbutton

self.foods\_dict[food\_item].var = IntVar()

# Set the variable parameter of the checkbutton

self.foods\_dict[food\_item]['variable'] = self.foods\_dict[food\_item].var

# Arrange the checkbutton in the window

self.foods\_dict[food\_item].pack(anchor='w')

payment\_label = Label(self, text="How do you want to pay?", bd=10)

payment\_label.pack(anchor='w')

# Create integer variable

self.var = IntVar()

self.var.set(0) # Use set() initialize the variable

self.payment\_methods = ["KBZpay", "Credit Card", "Other"]

for i, method in enumerate(self.payment\_methods):

self.pymt\_method = Radiobutton(self, text=method, variable=self.var, value=i)

self.pymt\_method.pack(anchor='w')

# Use ttk to add styling to button

style = ttk.Style()

style.configure('TButton', bg='skyblue', fg='white')

# Create button that will call the method to display text and

# close the program

next\_button = ttk.Button(self, text="Next", command=self.printResults)

next\_button.pack()

def printResults(self):

"""Print the results of the checkboxes and radio buttons."""

for cb in self.foods\_dict.values():

if cb.var.get():

print('Item selected: {}'.format(cb['text']))

index = self.var.get()

print("Payment method: {}".format(self.payment\_methods[index]))

self.quit()

if \_\_name\_\_ == "\_\_main\_\_":

app = SimpleGUI()

app.mainloop()

**Eg. 3**

# Python program to create a

# GUI report card using tkinter

# Import tkinter as tk

import tkinter as tk

# creating a new tkinter window

master = tk.Tk()

# assigning a title

master.title("Report Card")

# specifying geometry for window size

master.geometry("700x250")

# declaring objects for entering data

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)

# function to display the total subject

# credits total credits and SGPA according

# to grades entered

def display():

# Variable to store total marks

tot = 0

# 10\*number of subject credits

# give total credits for grade A

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

# grid method is used for placing

# the widgets at respective positions

# in table like structure .

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

tot += 40

# 9\*number of subject credits give

# total credits for grade B

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

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

tot += 36

# 8\*number of subject credits give

# total credits for grade C

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

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

tot += 32

# 7\*number of subject credits

# give total credits for grade D

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

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

tot += 28

# 6\*number of subject credits give

# total credits for grade P

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

tk.Label(master, text="24").grid(row=3, column=4)

tot += 24

# 0\*number of subject credits give

# total credits for grade F

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

tk.Label(master, text="0").grid(row=3, column=4)

tot += 0

# Similarly doing with other objects

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

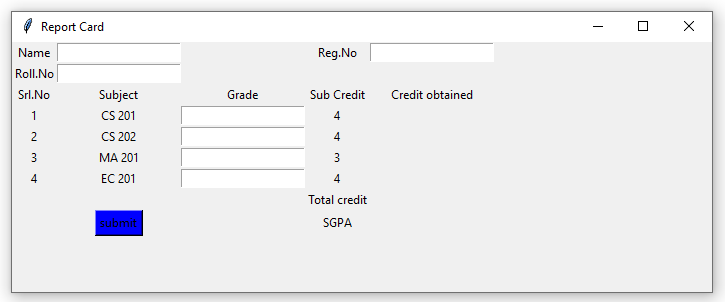
button1 = tk.Button(master, text="submit", bg="blue", command=display)

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)

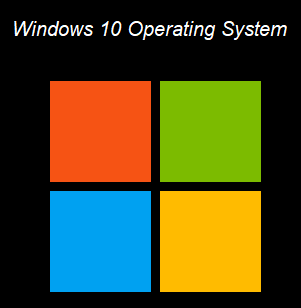
master.mainloop()



**Code Explanation:**

1. The code starts by creating a new tkinter window.
2. The title of the window is “Report Card.”
3. The geometry of the window is set to be 700×250 pixels.
4. Next, four objects are created: an entry object for holding data, an e1 object for holding student information, an e2 object for holding teacher information, and an e3 object for holding test information.
5. Each of these objects has a corresponding variable assigned to it.
6. For example, the entry object has a variable named e1 .
7. This variable stores the data entered into the window.
8. Similarly, each of the other objects also has a corresponding variable.
9. For example, the e2 object stores student information, and its corresponding variable is named s1 .
10. The same goes for the other variables in each of these objects.
11. Now that all of the variables have been initialized, some code is executed to create buttons and labels onscreen.
12. A button named btnStart is created and given a label of “Start.”
13. A label called lblMarkName is also created and given a text value of “Mark Sheet.”
14. Finally, two labels are created: one called lblStudentName and another called lblTeacherName .
15. These labels will display
16. The code creates a new tkinter window and assigns it the title “MARKSHEET”.
17. The geometry for the window is set to be 700×250.
18. Next, four objects are created which will be used to store data in the GUI.
19. These objects are named e1, e2, e3, and e4.
20. Finally, seven entries are created in each of these objects which will hold data about the students in the class.
21. The code starts by storing a variable, tot, to hold the total number of credits for each grade.
22. The code then loops through each of the grades in the data set and calculates the total number of credits for that grade.
23. For grades A, B, and C, the code adds 40, 36, and 32 respectively to tot.
24. For grades D and P, the code adds 24 and 28 respectively to tot.
25. Finally, for grades F and G (which have no subject credit), tot is set to 0.
26. The next part of the code sets up labels for each column in a table-like structure called a grid.
27. The labels are created using tk.Label() function calls with appropriate text values assigned as arguments.
28. The grid() method places these labels at respective positions in the table using row=row number/column=column number coordinates (e.g., tk.Label(master,”40″) will be placed at position (4, 4) in the table).
29. When all of the labels have been created, Tot is incremented by 40 for grades A-D and by 36 for grades E-P. Next, if e5 is equal to “A”, “B”, or “C”, then instead
30. The code displays the total subject credits and SGPA according to grades entered.
31. The code starts by storing the total marks in a variable called tot.
32. Next, it calculates the total credits for grade A by adding 40 to tot.
33. Similarly, it calculates the total credits for grade B, 36, and so on.
34. When it comes to grades below A, the code uses grid() method to place the widgets at respective positions in a table like structure.
35. For grades above A, there is no need to use grid() method as all marks are displayed in one line.
36. The last part of the code checks if any of the variables e5, e6 have values that match those of “A”.
37. If they do, then the corresponding label is
38. The code starts with a few basic labels for the different parts of the user interface.
39. The first label is “master”.
40. This is the main window that contains all of the other labels and controls.
41. Next, there are three labels for grades: CS 201, CS 202, and MA 201.
42. These labels will show the student’s current grade in each course.
43. The next set of labels shows subject credits: Sub Credit, 4, 4, and 3.
44. These labels will show how many credit hours have been earned in each subject area.
45. Note that these values are rounded to two decimal places (e.g., Sub Credit = .5).
46. Finally, there are four labels for credit obtained: Credit obtained, 1/2 Credit obtained, 3/4 Credit obtained, and Full Credit obtained.
47. These labels will show how much credit has been earned in each subject area so far (rounded to two decimal places).
48. The code creates a grid with six rows and six columns.
49. The first row contains three labels: CS 201, CS 202, and MA 201.
50. The second row contains four labels: EC 201, Grade, Sub Credit, and Credit obtained.
51. The third row contains two labels: 4 and 3.
52. The fourth row contains one label: 4.
53. The fifth row contains one label: 3.
54. The sixth row has no labels.
55. The code also creates a grid with six rows and six columns that displays the grades for the subject credits in alphabetical order.

Eg. 4



import turtle

turtle.bgcolor('black')

turtle.colormode(255)

def square():

for i in range (4):

turtle.forward(100)

turtle.right(90)

#Display Text

turtle.penup()

turtle.goto(90,100)

turtle.color("white")

style = ("Arial", 15, "italic")

turtle.write("Windows 10 Operating System", font=style, align="center")

#yellow box

turtle.left(90)

turtle.penup()

turtle.goto(100,-150)

turtle.pendown()

turtle.color('yellow')

turtle.begin\_fill()

square()

turtle.end\_fill()

#green box

turtle.penup()

turtle.goto(100,-40)

turtle.pendown()

turtle.color('green')

turtle.begin\_fill()

square()

turtle.end\_fill()

#red box

turtle.penup()

turtle.goto(-10,-40)

turtle.pendown()

turtle.color('red')

turtle.begin\_fill()

square()

turtle.end\_fill()

#blue box

turtle.penup()

turtle.goto(-10,-150)

turtle.pendown()

turtle.color('blue')

turtle.begin\_fill()

square()

turtle.end\_fill()

turtle.hideturtle()

turtle.done()