* Tkinter *

* BASIC PROGRAM *

Q.1 WAP in GUI to import Tkinter package & create a window and set its title.

→

```
From tkinter import *
win = TK()
win.title("welcome")
win.mainloop()
```

Q.2 set its title and a lable to the window.

→

```
From tkinter import *
win = Tk()
win.title("Document")

lbluserName = Label(win, text="User Name")
lblUserName.place(x=10, y=10)
name = stringVar()
entUserName = Entry(win, bg="red", fg="white", bd=10,
                    textvariable = name)
entUserName.place(x=80, y=10)
win.mainloop()
```

Q.3 WAP to create label & change the label font style

→

```
from
import tkinter import *
parent = tk.Tk()
parent.title("Document")
```

```
my_label = tk.label(parent, text="Hello", Font=("Arial Bold", k)
my_label.grid(column=0, row=0)
parent.mainloop()
```

Q.4 Set a default window size

→

```
import tkinter as tk
parent = tk.Tk()
parent.title("welcome")
parent.geometry('600 x 3000')
parent.mainloop()
```

Q.5 Disable to resize the window.

→

```
From tkinter import*
parent = Tk()
parent.title("welcome")
parent.resizeable(0,0)
parent.mainloop()
```

Q.6 ~~WAP to~~ Add labels & button

→

```
From tkinter import*
parent = Tk()
parent.title("welcome")
label = Label(parent, text="Username")
label.pack()
name = StringVar()
entUserName = Entry(parent, bg="red", fg="white",
                    textvariable = name)
```

```
entUserName.place(x=80, y=10)
btn = Button(win, text="Submit", bg="Green", command=sho
btn.place(x=10, y=80)
def show():
    x=name.get()
    print(x)
    lblmsg.config(text=x)
win.mainloop()
```

Q.7 WAP that implements event handling for button clicks.

→

```
import tkinter as tk
root = tk.Tk()
root.title("welcome")
label = tk.label(root, text="click the button & check the messag
label.pack()

def on_button_click():
    label.config(text="Button Clicked!")
button = tk.button(root, text="click me", command=on_button_cli
button.pack()
root.mainloop()
```

Q.8 Creates a basic menu bar with menu items.

→

```
import tkinter as tk
from tkinter import Menu
parent = tk.Tk()
parent.title("welcome")
```

```
menu_bar = Menu(parent)
parent.config(menu = menu_bar)

file_menu = Menu(menu_bar, tearoff = 0)
menu_bar.add_cascade(label="File", menu = file_menu)

edit_menu.add_command (label = "Cut")
edit_menu.add_command(label = "Copy")

help_menu = Menu(menu_bar, tearoff = 0)
menu_bar.add_cascade(label="Help", menu=help_menu)
help_menu.add_command(label = "About Spyder")
parent.mainloop()
```

Q.9 display message in a messagebox.

→

```
import tkinter as tk
from tkinter import messagebox
parent = tk.Tk()
parent.title("Messagebox")

def show_message():
    messagebox.showinfo("message", "Hello!")

button = tk.button(parent, text = "show Message",
        command = show_message)
button.pack()
parent.mainloop()
```

Q.10 Customize the appearance of label and button.

→

```
import tkinter as tk
parent = tk.Tk()
parent.title("welcome")

label = tk.label(parent, text = "label", font = ("Arial", 18),
                 fg = "white", bg = "red")
label.pack(pady = 10)
button = tk.button(parent, text = "button", font = ("Helvetica",
                   fg = "white", bg = "blue")
button.pack(pady = 10)
parent.mainloop()
```

Q.11 Create a window with a specific background color

→

```
import tkinter as tk
parent = tk.Tk()
parent.title("welcome")

parent.configure(bg = "lightpink")
parent.mainloop()
```

Q.12 ADD a image

→

```
import tkinter as tk
parent = tk.Tk()
parent.title("welcome")
```

```
image = Image.open("w3r_logo.png")
image = ImageTk.PhotoImage(image)
image_label = tk.label(parent, image = image)
image_label.pack()
parent.mainloop()
```

Q.13 Design a sample Calculater application

→

```
import tkinter as tk
def update_display(value):
    current_text = display_var.get()
    if current_text == "0":
        display_var.set(value)
    else
        display_var.set(current_text + value)

def clear_display():
    display_var.set("0")

def Calculate_result():
    try:
        result = eval(display_var.get())
        display_var.set(result)
    except Exception as e:
        display_var.set("Error")
parent = tk.TK()
parent.title("Calculator")

display_var = tk.stringVar()
display_var.set("0")
```

```
display_label = tk.Label(parent, textvariable=display_var,
                         font=("Arival",24), anchor="e",
                         bg="lightgray", padx=10, pady=10)
display_label.grid(row=0, column=0, columnspan=4)

button_layout = [
("7",1,0), ("8",1,1), ("9",1,2), ("/",1,3),
("4",2,0), ("5",2,1), ("6",2,2), ("*",2,3),
("1",3,0), ("2",3,1), ("3",3,2), ("-",3,3),
("0",4,0), (".",4,1), ("=",4,2), ("+",4,3),
]

for (text, row, col) in button_layout:
    button = tk.Button(parent, text=text, padx=10, pady=20
                       font=("Arial",18), command=lambda t=text:
                       update_display(t) if t != "=" else calculate_resu
    button.grid(row=row, column=col)

clear_button = tk.Button(parent, text="C", padx=1
                         pady=20, font=("Arival",18), command=
                                                      clear_display()
clear_button.grid(row=5, column=0, columnspan=3
parent.mainloop()
```

Q.14 TKinter based digital clock that display current tir

→

```
import tkinter as tk
import time

def update_time():
    current_time = time.strftime('%H:%M:%S')
    clock_label.config(text = current_time)
    root.after(1000, update_time)
root = tk.Tk()
root.title("Digital Clock")

clock_label = tk.Label(root, text = "", font = ("Helvetica",4
clock_label.pack(padx = 20, pady = 20)
update_time()
root.mainloop()
```

Q.15 Temperature Converter application.

→

PAGE:
DATE: / /

Q.16 login form with input fields for userid & password

→

```
import tkinter as tk
From tkinter import messagebox

def validate_login():
    userid = username_entry.get()
    password = password_entry.get()

    if userid == "admin" and password == "password":
        message.showinfo("Login successful")
    else:
        message.showinfo("Login failed")
parent = tk.Tk()
parent.titlel("Login form")

username_label = tk.label(parent, text = "Userid:")
username_label.pack()
username_entry = tk.Entry(parent)
username_entry.pack()

password_label = tk.label(parent, text = "Password:")
password_label.pack()
password.label= tk.Entry(parent, show = "*")
password.label.pack()

login_button = tk.button(parent, text="Login", Command =
login_button.pack()                              validate_login)
parent.mainloop()
```

Q.17 ADD tooltips to button & labels.

→

```
import tkinter as tk

def show_tooltips(event, tooltips_text):
    tooltips.geometry(f"+{event.x_parent+10}+{event.y_
                      event_parent+10}")
    tooltips_label.config(text = tooltips_text)
    tooltips.deiconify()

def hide_tooltips(event):
    tooltips.withdraw()

parent = tk.Tk()
parent.title("Tooltips")

button = tk.button(parent, text = "button")
button.pack(padx = 10, pady = 10)
button.bind("<Enter>", lambda event, text = "This is a
        button": show_tooltips(event, text))
button.bind("<leave>", hide_tooltips)

label = tk.label(parent, text = "label")
label.pack(padx = 10, pady = 10)
label.bind("<Enter>", lambda event, text = "This is a
        label": show_tooltips(event, text))
label.bind("<leave>", hide_tooltips)

tooltips = tk.Toplevel(parent)
tooltips.withdraw()
```

```
tooltips_label = tk.label (tooltips, text = '', background =
                 "lightyellow", relief = "solid", borderwidth=1)
tooltips_label.pack()
parent.mainloop()
```

Q.18 Create a window that closes when a "close" button is clicked.

→
```
import tkinter as tk

def close_window():
    parent.destroy()

parent = tk.Tk()
parent.title ("Close window")

label = tk.label (parent, text = "Click me")
label.pack()

close_button = tk.button (parent, text = "Close",
                          command = close_window)
Close_button.pack()
parent.mainloop()
```

Q.19 Create tkinter application that allows user to select & display their specified Favourite Color

→
```
import tkinter as tk
from tkcolorpicker import askcolor
```

```
def choose_color():
    color = askcolor()[1]
    if color:
        color_label.config(text=f"selected color:{color}",
                           bg=color)

parent = tk.Tk
parent.title("Color")

color_label = tk.label(parent, text="selected color: None",
                       font("Helvetica", 14), padx=10, pady=10)
color_label.pack()

choose_button = tk.button(parent, text="choose color",
                          command=":choose_color)
choose_button.pack(pady=10)
parent.mainloop()
```

20 TKinter based timer application that counts down from a specified time when started.

→

```
import tkinter as tk
from tkinter import ttk

def start_timer():
    global remaining_time
    try:
        minutes = int(minutes_entry.get())
        seconds = int(seconds_entry.get())
        remaining_time = minutes * 60 + seconds
```

```
        update_timer()
        start_button.config(state="disabled")
        stop_button.config(state="active")
    except ValueError:
        pass

def update_timer():
    global remaining_time
    if remaining_time > 0:
        minutes = remaining_time // 60
        seconds = remaining_time % 60
        timer_label.config(text=f"{minutes:02d}:{seconds:
        remaining_time -= 1
        parent.after(1000, update_timer)
    else:
        timer_label.config(text="00:00")
        start_button.config(state="active")
        stop_button.config(state="disabled")

def stop_timer():
    global remaining_time
    remaining_time = 0
    timer_label.config(text="00:00")
    start_button.config(state="active")
    stop_button.config(state="disabled")

parent = tk.Tk()
parent.title("Timer Application")
```

```
minute_label = tK.label(parent , text = "Minutes:")
minute_label.pack()
minute_labelentry = tK.Entry(parent)
minute_entry.grid (row) = 0, Column = 1)
minute_entry.insert (0, "0")
seconds_label = tK.label(parent, text = "Seconds:")
seconds.label.grid (row) = 1, Column = 0)
seconds_entry = tK.Entry(parent)
seconds_entry.grid (row) = 1, Column = 0)
second_entry.insert (0, "0")

timer_label = tK.label(parent, text = "00:00", Font = ("Helvetica", 48)
timer_label.grid (row) = 2, Columnspan = 2)

start_button = ttK.button(parent, text = start,
                          Command = start_timer)
start_button.grid (row) = 3, Column = 0)

stop_button = ttK.button(parent , text = stop, state = "disable"
                         Command = stop_timer)
stop_button.grid(row) = 3, Column = 1)

remaining_time = 0
parent.mainloop()
```

PAGE:
DATE: / /

Q.1 ADD button in a Application.

→

```
import tkinter as tk
parent = tk.Tk()
parent.title("welcome")
my_button = tk.button(parent, text='Quit', height=1,
                      width=35, command=parent.destroy)
my_button.pack()
parent.mainloop()
```

Q.2 WAP to add Canvas in your application

→

```
import tkinter as tk
parent = tk.Tk()

Canvas_width = 100
Canvas_height = 80
w = tk.Canvas(parent, width=Canvas_width, height=
                                        Canvas_height)
w.pack

y=int(canvas_height/2)
w.create_line(0,y, canvas_width,y, Fill="#476042")
parent.mainloop()
```

Q.3 Create two button exit & hello

→

```
import tkinter as tk

def write_text():
    print("TKinter is easy to create GUI")

parent = tk.Tk()
Frame = tk.Frame(parent)
frame.pack()

text_disp = tk.button(Frame, text="Hello", Command=write_te
text_disp.pack (side = tk.LEFT)

exit_button = tk.button(frame, text='Exit', fg=green, Command=
exit_button.pack (side = tk.RIGHT)                    quit)
parent.mainloop()
```

Q.4 Create Combobox with three option.

→

```
import tkinter as tk
From tkinter import ttk
root = tk.Tk()
my_str_var = tk.StringVar()

my_combobox = ttk.Combobox (root, textvariable=my_str
                            ,values=("PHP", "JAVA", "Python
my_combobox.pack()
root.mainloop()
```

PAGE:
DATE: / /

Q.5 Create a Checkbutton widget.

→
```
import tkinter as tk
From tkinter import ttk

root = tk.Tk()
my_boolean_var = tk.BooleanVar()

my_checkbutton = ttk.Checkbutton(text="check",
                                 variable=my_boolean_var)
my_checkbutton.pack()
root.mainloop()
```

Q.6 Create a spinbox widget.

→
```
import tkinter as tk
root = tk.Tk()
text_var = tk.DoubleVar()

spin_box = tk.Spinbox(root, from=0.6, to=50.0, increm
                      =.01, textVariable=text_var)
spin_box.pack()
root.mainloop()
```

Q.7 Create a text widget.

→
```
import tkinter as tk
root = tk.Tk()
mytext = tk.Text(root)

mytext.insert('1.0', "- Python")
mytext.insert('1.19', 'Practice.')
```

```
mytext.delete('1.0')
mytext.delete('end - 2 chars')
mytext.pack()
root.mainloop()
```

Q.8 Create three single line text-box

→

```
import tkinter as tk
parent = tk.Tk()
parent.geometry("400x250")

name = tk.label(parent, text="Name").place(x=30, y=50)
email = tk.label(parent, text="User Id").place(x=30, y=90)
password = tk.label(parent, text="password").place(x=30, y=130)
Submitbtn = tk.button(parent, text="Submit").place(x=120, y=17

entry1 = tk.Entry(parent).place(x=85, y=50)
entry2 = tk.Entry(parent).place(x=85, y=90)
entry3 = tk.Entry(parent).place(x=90, y=130)
parent.mainloop()
```

Q.9 Create three radio button widgets.

→

```
import tkinter as tk
parent = tk.Tk()
parent.title("Radiobutton")
parent.geometry('350x200')

radio1 = tk.Radiobutton(parent, text='first', value=1)
```

```
radio2 = tk.Radiobutton(parent, text="second", value=2)
radio3 = tk.Radiobutton(parent, text="third", value=3)
radio1.grid(column=0, row=0)
radio2.grid(column=1, row=0)
radio3.grid(column=2, row=0)
parent.mainloop()
```

Q. 10 Create a scrolled Text widget.

→

```
import tkinter as tk
import tkinter.scrolltext as tkst
parent = tk.Tk()
parent.title("scrolltext")
parent.geometry('350x200')
txt = tkst.ScrolledText(parent, width=40, height=10)
txt.grid(column=0, row=0)
parent.mainloop()
```

Q. 11 Create a progess bar. widget

→

```
import tkinter as tk
import tkinter.scrolltext as tkst
from tkinter.ttk import progessbar
from tkinter import ttk
parent = tk.Tk()
parent.geometry("350 x 200")
style = ttk.Style()
style.theme_use('default')
style.configure("black.Horizontal.TProgessbar", bg="gree
```

```
bar = progessbar(parent, length=220, style='black.horizontal.
bar['value'] = 80                                  Tprogressbar')
bar.grid(column=0, row=0)
parent.mainloop()
```

Q.12 Create a Listbox bar widgets.

→

```
import tkinter as tk
parent = tk.Tk()
parent.geometry("250x200")
label1 = tk.label(parent, text="languages"
listbox = tk.Listbox(parent)
listbox.insert(1, "PHP")
listbox.insert(2, "JAVA")
listbox.insert(3, 'C')
listbox.insert(4, 'C++')
label1.pack()
listbox.pack()
parent.mainloop()
```

Q.13 Create a notebook with three ~~labels~~ taps.

→

```
import tkinter as tk
From tkinter import ttk

def create_tab1(tab):
    label = ttk.Label(tab, text="Java")
    label.pack(padx=20, pady=20)
```

```
def create_tab2(tab):
    label = ttk.label(tab, text = "Python")
    label.pack(padx = 20, pady = 20)

def create_tab3(tab):
    label = ttk.label(tab, text = "C++")
    label.pack(padx = 20, pady = 20)

def main():
    parent = tk.Tk()
    parent.title("Tabbed Interface")
    notebook = ttk.Notebook(parent)

    tab1 = ttk.Frame(notebook)
    tab2 = ttk.Frame(notebook)
    tab3 = ttk.Frame(notebook)
    notebook.add(tab1, text = "Java")
    notebook.add(tab2, text = "Python")
    notebook.add(tab3, text = "C++")
    create_tab1(tab1)
    create_tab2(tab2)
    create_tab3(tab3)
    notebook.pack(padx = 10, pady = 10, fill = "both", expand = Tru
    parent.mainloop()

if __name__ == "__main__":
    main()
```

PAGE:
DATE: / /

Q.4 Create a Treeview widget

→

```
import tkinter as tk
from tkinter import ttk

def add_child_item():
    selected_item = tree.selection()
    if selected_item:
        parent_item = selected_item[0]
        tree.insert(parent_item, "end", text="child item")

def delete_item():
    selected_item = tree.selection()
    if selected_item:
        tree.delete(selected_item)

def add_sibling_item():
    selected_item = tree.selection()
    if selected_item:
        parent_item = tree.parent(selected_item[0])
        tree.insert(parent_item, "end", text="sibling Item")

parent = tk.Tk()
parent.title("Treeview")
tree = ttk.Treeview(parent)
tree.pack(padx=10, pady=10, fill="both", expand=True)
parent_item = tree.insert("", "end", text="parent Item")

add_child_button = tk.button(parent, text="ADD child item",
                             command=add_child_item)
```

```
add_child_button.pack()

add_sibling_button = tk.button(parent, text="Add Sibling
                    Item", Command = add_sibling_item)
add_sibling_button.pack()

delete_button = tk.button(parent, text="Delete item",
                    command = delete_item)
delete_button.pack()
parent.mainloop()
```

Q.15 Create a Menu bar with File, edit & help menu, each containing submenu items.

→

```
import tkinter as tk
From tkinter import Menu

def new_file():
    print("New file")
def open_File():
    print("open file")
def save_File():
    print("save file")
def cut_text():
    print("Cut Texts")
def copy_text():
    print("Copy Texts")
def paste_text():
    print("Paste text")
```

```
def about():
    print("About this Application")
parent = tk.Tk()
menu_bar = Menu
```