

Chapter 15

A. Multiple Choice Answers

1. B
2. B
3. C
4. A
5. D
6. A
7. 4
8. B
9. A
10. A

B. Exercise Questions

1. Recall how to create label with text "I Love python Programming" in color red.

The syntax to create Label widget is as follows.

Solution:

Syntax:

```
Label(ref_parent_widget, text = "" , option = placeholder)
```

Thus, inorder to create label in red color, make use of option foreground i.e. **fg='red'** to appear text in red color.

Program:

```
import tkinter as TK      # Import all definitions from Tkinter
root = TK.Tk()            # Create drawing window
L1 = TK.Label(root, text = "I Love Python Programming",fg='red')
L1.pack()
root.mainloop()
```

2. Demonstrate with appropriate example use of grid geometry manager.

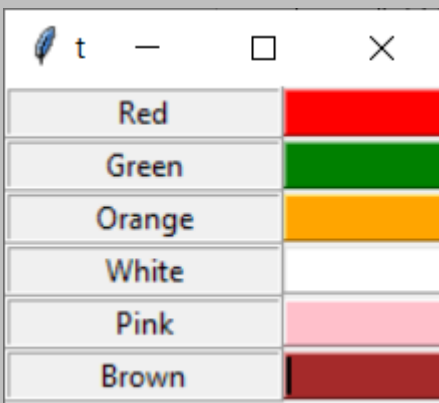
Grid manager is used to place the widget in appropriate row and column. The master widget is split into rows and columns and each part of the table i.e. cell can hold widget.

The following program make use of grid geometry manager, and display list of color name in one column and respective color in other column.

Program:

```
import tkinter as tk
colors = ['Red', 'Green', 'Orange', 'White', 'Pink', 'Brown']
r = 0
for clr in colors:
    tk.Label(text = clr, relief = tk.RIDGE,
             width=15).grid(row = r, column= 0)
    tk.Entry(bg = clr, relief = tk.RIDGE,
            width=10).grid(row = r, column = 1)
    r = r + 1
tk.mainloop()
```

Output:



3. Specify, what exactly the difference between places, pack is and gird geometry manager.

Solution:

- Grid geometry manager arranges widgets along row and column boundaries i.e. it places widgets in a two dimensional grid. It is great for creating tables and other structured types of layouts.
- The pack geometry manager organizes widgets in horizontal and vertical boxes. The layout is controlled with the fill, expand, and side options.

- Place geometry manager make use of two positioning systems (absolute and relative) to position widget in their parent frame. Absolute positioning makes use of actual x and y coordinates. Whereas, relative positioning positions widget as a percentage relative to the size of the parent frame or root window.

4. Express how to make use of listbox widget.

Listbox contains, list of items in an application form, from which user can make a selection. The syntax to create listbox widget is as follows.

Syntax:

```
listbox(ref_parent_window, *options)
```

The following program demonstrate the use of "List Box" Widget containing list of cities. On click of any one of the city it will display the selected city.

Program:

```
from tkinter import Tk, BOTH, Listbox, StringVar, END
from tkinter.ttk import Frame, Label
class Example(Frame):
    def __init__(self):
        super().__init__()
        self.initUI()

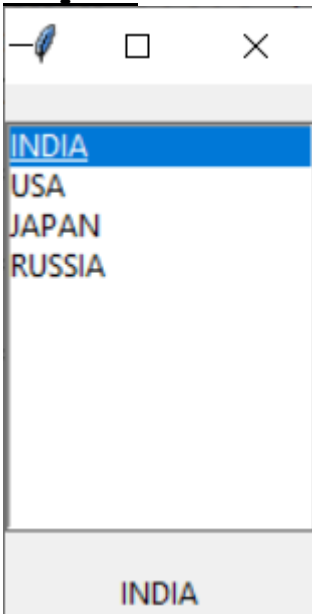
    def initUI(self):
        self.master.title("Listbox")
        self.pack(fill=BOTH, expand=1)
        Country = ['INDIA', 'USA',
                   'JAPAN', 'RUSSIA']

        lb = Listbox(self)

        for i in Country:
            lb.insert(END, i)

        lb.bind("<<ListboxSelect>>", self.onSelect)
        lb.pack(pady=15)
        self.var = StringVar()
        self.label = Label(self, text=0, textvariable=self.var)
        self.label.pack()
```

```
def onSelect(self, val):  
    sender = val.widget  
    idx = sender.curselection()  
    value = sender.get(idx)  
    self.var.set(value)  
  
def main():  
    root = Tk()  
    Example()  
    root.mainloop()  
  
if __name__ == '__main__':  
    main()
```

Output:

5. Illustrate with example use of command option within the button widget.

The button widget can be created by using the Button class. The syntax to create Button widget is as follows:

Syntax:

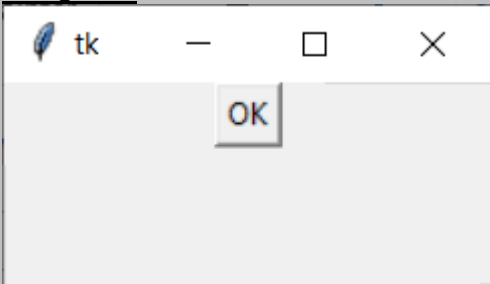
```
Button(ref_parent_widget, option = placeholder)
```

```
import tkinter as TK
```

```
class ButtonEventHandler:
    def __init__(self):
        root = TK.Tk()
        self.butOk = TK.Button(root, text = "OK", command =
self.functionOk)
        self.butOk.pack()
        root.mainloop()

    def functionOk(self):
        print("OK Button is clicked")

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

Output:

#On click of "OK" Button
OK Button is clicked

C. Programming Assignments

1.

```
import tkinter as tk
from tkinter import Label, Entry, Button, Text, END

class PersonalInfo:
    def __init__(self):
        root = tk.Tk()
        root.resizable(width='false', height='false')
        root.configure(background='white')
        lable_one = Label(root, text="First Name",
                           font="Times 15 bold",
                           fg='Black', bg="WHITE")

        # Add label to main window
        lable_one.grid(row=1, column=1)

        # Create a entry field
```

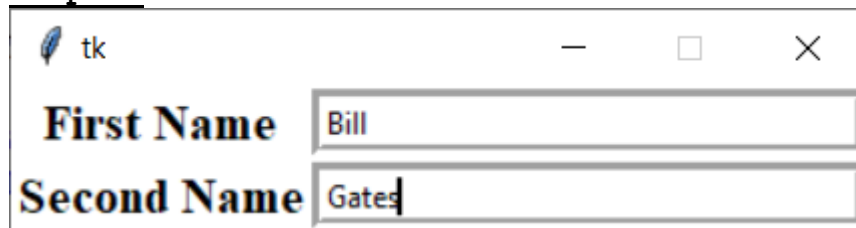
```
self.entry1 = Entry(root, bd=5,
                    relief='groove', width=35)
# add entry field to main window
self.entry1.grid(row=1, column=2)

label_two = Label(root, text="Second Name",
                  font="Times 15 bold",
                  fg='Black', bg="WHITE")
label_two.grid(row=2, column=1)

self.entry2 = Entry(root, bd=5,
                    relief='groove', width=35)
# add entry field to main window
self.entry2.grid(row=2, column=2)
root.mainloop()
```

PersonalInfo()

Output:



2.

```
import tkinter as tk
from tkinter import Label, Entry, Button, Text, END

class PersonalInfo:
    def __init__(self):
        root = tk.Tk()
        root.resizable(width='false', height='false')
        root.configure(background='white')
        lable_one = Label(root, text="First Name",
                          font="Times 15 bold",
                          fg='Black', bg="WHITE")
        # Add label to main window
        lable_one.grid(row=1, column=1)
        # Create a entry field
        self.entry1 = Entry(root, bd=5,
                            relief='groove', width=35)
        # add entry field to main window
```

```
self.entry1.grid(row=1, column=2)

label_two = Label(root, text="Second Name",
                  font="Times 15 bold",
                  fg='Black', bg="WHITE")
label_two.grid(row=2, column=1)

self.entry2 = Entry(root, bd=5,
                   relief='groove', width=35)
# add entry field to main window
self.entry2.grid(row=2, column=2)

btn = Button(root, text="Submit", fg="black",
             font="Times 15 bold",
             bg="Gray", command=self.Submit)
btn.grid(row=5, column=2)

btn.grid(row=4, column=2)
root.mainloop()

def Submit(self):
    fname = (self.entry1.get())
    sname = (self.entry2.get())
    print('First Name:', fname)
    print('Second Name:', sname)
PersonalInfo()
```

Output:

The screenshot shows a Tkinter window titled "tk" with a standard macOS-style title bar (red, yellow, green buttons). Inside the window, there are two text entry fields. The first field is preceded by the label "First Name" in a bold, black, serif font. The field contains the text "Virat". The second field is preceded by the label "Second Name" in the same font. The field contains the text "Kohli". Below the second field is a gray button with the text "Submit" in a bold, black, serif font.

#On Click of Submit Button

First Name: Virat
Second Name: Kohli

3.

```
import tkinter as tk
from tkinter import Label, Entry, Button, Text, END

class Addition:
    def __init__(self):
        root = tk.Tk()
        root.resizable(width='false', height='false')
        root.configure(background='white')
        label_one = Label(root, text="First No",
                           font="Times 15 bold",
                           fg='Black', bg="WHITE")
        # Add label to main window
        label_one.grid(row=1, column=1)
        # Create a entry field
        self.entry1 = Entry(root, bd=5,
                             relief='groove', width=35)
        # add entry field to main window
        self.entry1.grid(row=1, column=2)

        label_two = Label(root, text="Second Number",
                           font="Times 15 bold",
                           fg='Black', bg="WHITE")
        label_two.grid(row=2, column=1)

        self.entry2 = Entry(root, bd=5,
                             relief='groove', width=35)
        # add entry field to main window
        self.entry2.grid(row=2, column=2)

        label_three = Label(root, text="Sum",
                             font="Times 15 bold",
                             fg='Black', bg="WHITE")
        label_three.grid(row=3, column=1)

        self.entry3 = Entry(root, bd=5,
                             relief='groove', width=35)
        # add entry field to main window
        self.entry3.grid(row=3, column=2)
```



```
btn = Button(root, text="Addition", fg="black",
              font="Times 15 bold",
              bg="Gray", command=self.add_numbers)
btn.grid(row=5, column=2)

btn = Button(root, text="Clear", fg="black",
              font="Times 15 bold",
              bg="Gray", command=self.clear)

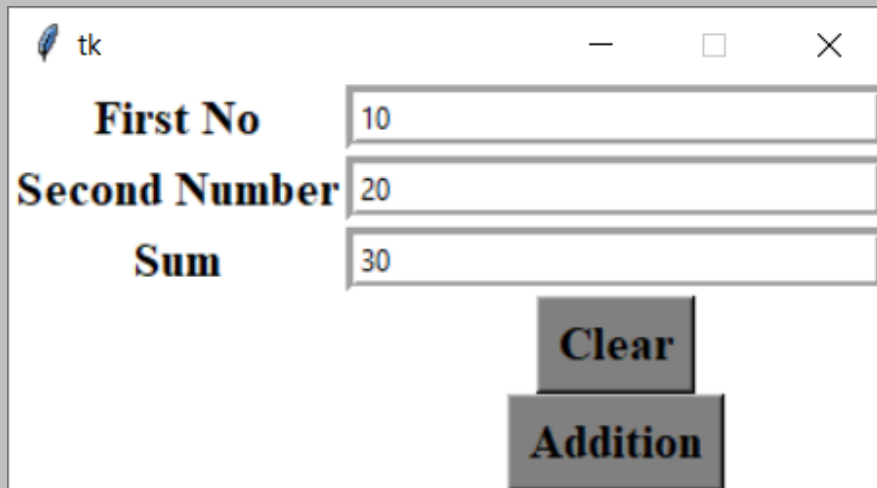
btn.grid(row=4, column=2)
root.mainloop()

def add_numbers(self):
    fno = int(self.entry1.get())
    sno = int(self.entry2.get())
    self.entry3.delete(0,END)
    self.entry3.insert(0, (fno+sno))
    print(fno+sno)

def clear(self):
    self.entry1.delete(0,END)
    self.entry2.delete(0,END)
    self.entry3.delete(0,END)
```

Addition()

Output:



The screenshot shows a Tkinter window titled "tk" with a standard macOS-style title bar (red, yellow, green buttons). Inside the window, there are three text labels on the left: "First No", "Second Number", and "Sum". To the right of each label is a text entry field. The "First No" field contains the number "10", the "Second Number" field contains "20", and the "Sum" field contains "30". Below these fields, there are two buttons. The top button is labeled "Clear" and the bottom button is labeled "Addition". Both buttons have a gray background and black text.

```
import tkinter as TK
from tkinter import *

class RadioButtonDemo:
    def __init__(self):
        self.root = TK.Tk()
        self.lbl = Label(self.root,
                         text = "Please Select one Option:")
        self.var = TK.IntVar()
        self.c1 = TK.Radiobutton(self.root, text="Male",
                                variable=self.var,
                                value=1,command=self.get_value)

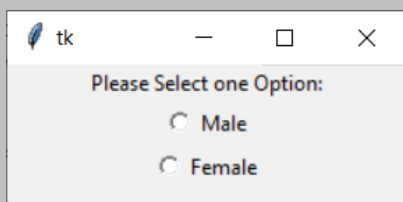
        self.c2 = TK.Radiobutton(self.root,
                                text="Female",
                                variable=self.var,
                                value=2,comman=self.get_value)

        # Packing buttons for display
        self.lbl.pack()
        self.c1.pack()
        self.c2.pack()

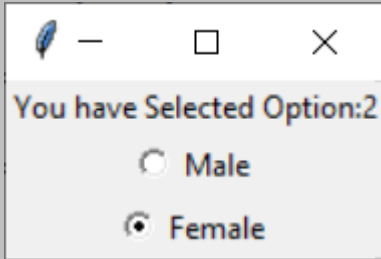
        # Displaying, and running mainloop
        self.root.mainloop()

    def get_value(self):
        selection = 'You have Selected Option:' \
                   + str(self.var.get())
        self.lbl.config(text = selection)

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

Output:

#After selecting one of the option



5.

```
import tkinter as TK
from tkinter import *

class CheckButtonDemo:
    def __init__(self):
        self.root = TK.Tk()
        self.is_check_butt_1 = IntVar()
        self.is_check_butt_2 = IntVar()

        self.check1 = Checkbutton(self.root, text="Crikcet",
                                   onvalue = 1, offvalue = 0,
                                   variable = self.is_check_butt_1,
                                   command = self.evaluate)

        self.check2 = Checkbutton(self.root, text="football",
                                   onvalue = 1,
                                   offvalue = 0,
                                   variable = self.is_check_butt_2,
                                   command = self.evaluate)

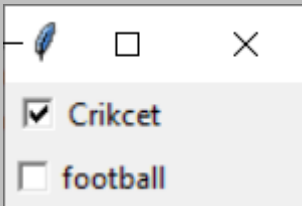
        self.check1.grid(row=1,column = 1)

        self.check2.grid(row=2,column = 1)

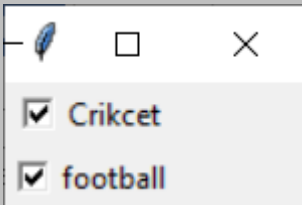
    def evaluate(self):
        if ((self.is_check_butt_1.get() == 1) and \
            (self.is_check_butt_2.get() == 0)):
            print("I Like Game of Cricket")
        elif((self.is_check_butt_1.get() == 0) and \
            (self.is_check_butt_2.get() == 1)):
            print("I Like Game of FootBall")
        else:
```

```
print("I Love Both")

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

Output:

```
>>> I Like Game of Cricket
```



```
>>> I Love Both
```