Module 13: Python GUI

tkinter:

- The tkinter is the module for
- To use tkinter we need to import Tkinter module:
  - import tkinter
  - Or
  - from tkinter import *
- It is part of the core Python modules and hence no installation required.
- Its is simple and cross-platform.
- It is also used in Ruby and Perl.

Method	Description
Tk()	To initialize Tkinter we need to create Tk root widget. It is an ordinary window with title bar and min. max and close button. We should have one root and it must be created before any other widgets.
mainloop()	The program will stay in the event loop until we close the window.
title()	It sets the title of the window.
geometry()	Its specifies the dimension and the placement of the window. width ✕ height + x + y
resizable()	It is for creating a fixed – width form.
iconbitmap()	It is for changing the windows icon.

Geometry Manager:

- It is the manager used for organizing the widgets before placing them in the parent widget.
- The three managers are:
  - pack() ➔
  - place() ➔
  - grid() ➔

The **Pack** geometry manager packs widgets in rows or columns.

fill:	It specifies whether the widget should occupy all the space provided by the master. If NONE (default) then keep the widgets original size. If X then fill horizontally If Y then fill vertically
width:	width of the button in letters.
height:	height of the button in text lines.
side:	it specifies which side to pack the widget. To pack widgets vertically use TOP(default.) To pack widgets horizontally use LEFT. We can also pack the widgets along BOTTOM and RIGHT edges.
padx:	→ It is for specifying the external padding of the x axis components. Similarly we have pady also. ipadx and ipady for internal padding.

The **Place** geometry managers allows us to explicitly set the position and size of the widgets.
x and y position of the X and Y Axis respectively.
width and height of the button respectively.

The **Grid** geometry manager puts the widgets in a 2-dimensional table.
The master widget is split into a number of rows and columns, and each "cell" in the resulting table can hold a widget.
Note that the column number defaults to 0 if not given.

row:	insert the widget at this row. Row number starts with 0. if omitted, it defaults to first empty row in the grid.
column:	➔ insert the widget at this column. Column number starts from 0. if omitted it defaults to 0.

Widgets:

- Button:
  - Button is a widget to implements buttons that contain text or images and we can associate a Python function with button which is called when button is pressed.
- messagebox:
  - It is used to display message boxes in applications.
  - 1) showinfo(windowtitle, text)
  - 2) showwarning(windowtitle, text)
  - 3) showerror(windowtitle, text)

---

tkinter Callbacks:

- Callback is Python code that is called by Tk when something happens.
- Button provides a command callback which is called when user clicks the button.
- For each function object, the Tkinter interface layer registers a Tk command with unique name. when that Tk command is called by the Button implementation, the command calls the corresponding python function.

Passing Arguments to Callbacks:

- tkinter's Button widget doesn't pass any information to the callback. This makes things a bit complicated if we want to use the same callback for several buttons.
- We can use lambda expression as a link between Tkinter and the callback function.

Module 13: Python tkinter

Q1) What is the result?

```
import tkinter
root = Tk()
root.mainloop()
```

Options:

A. It will create a window without close button.
B. It will create a window with all 3 buttons.
C. Error
D. It will create window but it wont be visible.

Solution:

Q2) What is p, q, r & s? **root.geometry("pxq+r+s")**

Options:

A. q = width, p = height, s = x and r = y
B. p = width, q = height, s = x and r = y
C. q = width, p = height, r = x and s = y
D. p = width, q = height, r = x and s = y

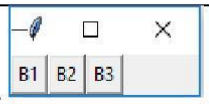
Solution:

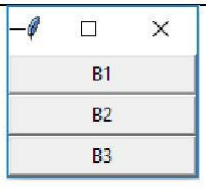
Q3) What is the result?

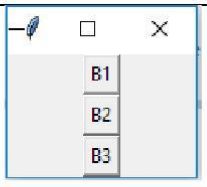
```
from tkinter import *
win = Tk()
b1 = Button(win, text="B1")
b2 = Button(win, text="B2")
b3 = Button(win, text="B3")
b1.pack()
b2.pack()
b3.pack()
win.mainloop()
```

Options:

A. Error

B. 

C. 

D. 

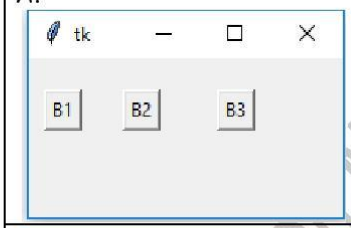
Solution:

Q4) What is the result?

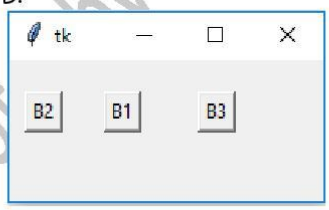
```
from tkinter import *
win = Tk()
b1 = Button(win, text="B2")
b2 = Button(win, text="B3")
b3 = Button(win, text="B1")
b1.place(x=10, y = 20)
b2.place(x = 60, y = 20)
b3.place(x = 120, y = 20)
win.mainloop()
```

Options:

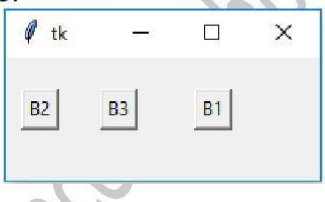
A.



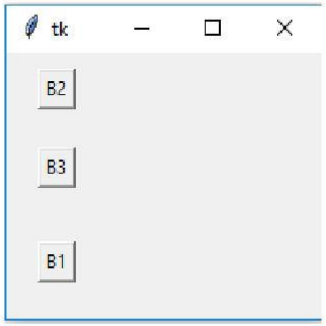
B.



C.



D.



Solution:

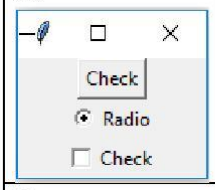
Q5) What is the result?

```
from tkinter import *
win = Tk()
b1 = Button(win, text="Radio")
r1 = Radiobutton(win, text='Button')
c1 = Checkbutton(win, text='Check')
b1.pack()
r1.pack()
c1.pack()
win.mainloop()
```

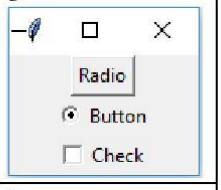
Solution:

Options:

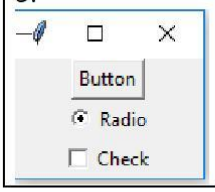
A.



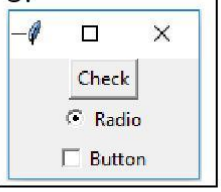
B.



C.



D.



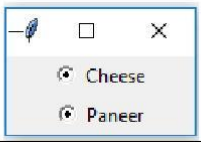
Q6) What is the result?

```
from tkinter import *
win = Tk()
choice = IntVar()
choice.set(1)
choice.set(2)

r1 = Radiobutton(win, text='Cheese',
 variable=choice, value=1)

r2 = Radiobutton(win, text='Paneer',
variable=choice, value=2)

r1.pack()
r2.pack()
win.mainloop()
```

Solution:

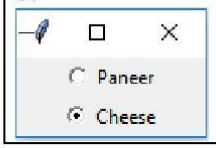
Options:

A.



B.

C.



D.