



Python

Experiment No. 11

Aim: To implement GUI Frame and Canvas application using Tkinter.

Problem Statements:

1. Create a GUI form with the details like (Name, age(take a password field to hide age), city, gender (radiobutton), favourite game (check button)Etc) and print the details on the frame area.
2. Create a GUI frame to find factorial of an input number. User is allowed to enter a number into the text field whose factorial is to be determined. On pressing the button the value of the text field is firstly converted into integer and then processed to find its factorial. The result will get displayed on command prompt.
3. Create a Christmas tree using various shapes (Triangle, rectangle(for stem), circle(for decoration), of in Canvas

Theory:

Tkinter Programming

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Frame Application:

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps -

- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

Tkinter Widgets

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

Different types of widgets are as follows:

Sr.No.	Operator & Description
1	Button The Button widget is used to display buttons in your application.
2	Canvas The Canvas widget is used to draw shapes, such as lines, ovals, polygons and

	rectangles, in your application.
3	Checkbutton The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.
4	Entry The Entry widget is used to display a single-line text field for accepting values from a user.
5	Frame The Frame widget is used as a container widget to organize other widgets.
6	Label The Label widget is used to provide a single-line caption for other widgets. It can also contain images.

Geometry Management

All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area. Tkinter exposes the following geometry manager classes: pack, grid, and place.

The pack() Method - This geometry manager organizes widgets in blocks before placing them in the parent widget.

The grid() Method - This geometry manager organizes widgets in a table-like structure in the parent widget.

The place() Method - This geometry manager organizes widgets by placing them in a specific position in the parent widget.

Canvas Application

The Canvas is a rectangular area intended for drawing pictures or other complex layouts. You can place graphics, text, widgets or frames on a Canvas.

Syntax:

Here is the simple syntax to create this widget –

```
w = Canvas ( master, option=value, ... )
```

Parameters

master – This represents the parent window.

options – Here is the list of most commonly used options for this widget. These options can be used as key-value pairs separated by commas.

Some of the options are as follows :

Sr.No.	Option & Description
1	bd Border width in pixels. Default is 2.
2	bg Normal background color.
3	confine If true (the default), the canvas cannot be scrolled outside of the scroll region.
4	cursor Cursor used in the canvas like <i>arrow</i> , <i>circle</i> , <i>dot etc.</i>
5	height Size of the canvas in the Y dimension.

Q.1 Create a GUI form with the details like (Name, age(take a password field to hide age), city, gender (radiobutton), favouritegame (check button)Etc) and print the details on the frame area.

Code:

```
import tkinter as tk

def print_details():
    name = name_entry.get()
    age = age_entry.get()
    city = city_entry.get()
    gender = gender_var.get()
    favorite_games = [game for game, var in
        favorite_games_vars.items() if var.get()]

    details_text.delete("1.0", tk.END)
    details_text.insert(tk.END, f"Name: {name}\n")
    details_text.insert(tk.END, f"Age: {age}\n")
    details_text.insert(tk.END, f"City: {city}\n")
    details_text.insert(tk.END, f"Gender: {gender}\n")
    details_text.insert(tk.END, f"Favorite Games: {'',
        '.join(favorite_games)}\n")

# Create the main window
window = tk.Tk()
window.title("GUI Form")
window.resizable(False, False) # Make window non-resizable

# Create labels and entry fields
name_label = tk.Label(window, text="Name:")
name_label.grid(row=0, column=0)
name_entry = tk.Entry(window)
name_entry.grid(row=0, column=1)

age_label = tk.Label(window, text="Age:")
age_label.grid(row=1, column=0)
age_entry = tk.Entry(window, show="*") # Password field for age
age_entry.grid(row=1, column=1)

city_label = tk.Label(window, text="City:")
city_label.grid(row=2, column=0)
city_entry = tk.Entry(window)
city_entry.grid(row=2, column=1)

# Create radio buttons for gender
```

```

gender_var = tk.StringVar(value="Male")
male_radio = tk.Radiobutton(window, text="Male",
variable=gender_var, value="Male")
male_radio.grid(row=3, column=0)
female_radio = tk.Radiobutton(window, text="Female",
variable=gender_var, value="Female")
female_radio.grid(row=3, column=1)

# Create check buttons for favorite games
favorite_games_vars = {
    "Cricket": tk.BooleanVar(value=False),
    "Football": tk.BooleanVar(value=False),
    "Chess": tk.BooleanVar(value=False)
}
for i, (game, var) in
enumerate(favorite_games_vars.items()):
    check_button = tk.Checkbutton(window, text=game,
variable=var)
    check_button.grid(row=4 + i, column=0, columnspan=2)

# Create button to print details
print_button = tk.Button(window, text="Print Details",
command=print_details)
print_button.grid(row=7, column=0, columnspan=2)

# Create text area to display details
details_text = tk.Text(window, height=5, width=30)
details_text.grid(row=8, column=0, columnspan=2)

# Start the GUI event loop
window.mainloop()

```

Output:

GUI Form

Name: Chaitanya Shah

Age: **

City: Mumbai

☒ Male ☐ Female

☒ Cricket

☒ Football

☒ Chess

Name: Chaitanya Shah
Age: 19
City: Mumbai
Gender: Male
Favorite Games: Cricket, Footb

Q.2 Create a GUI frame to find factorial of an input number. User is allowed to enter a number into the text field whose factorial is to be determined. On pressing the button the value of the text field is firstly converted into integer and then processed to find its factorial. The result will get displayed on command prompt.

Code:

```
import tkinter as tk
import math

def calculate_factorial():
    try:
        num = int(entry.get())
        if num < 0:
            result_label.config(text="Factorial is not
defined for negative numbers")
        else:
            factorial = math.factorial(num)
            result_label.config(text=f"Factorial of {num}
is: {factorial}")
    except ValueError:
        result_label.config(text="Invalid input. Please
enter an integer.")

# Create main window
window = tk.Tk()
window.title("Factorial Calculator")
window.geometry("200x100") # Set window size
window.resizable(False, False) # Make window non-resizable

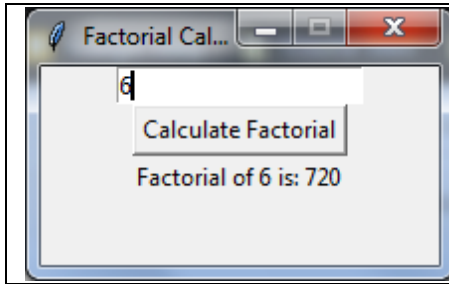
# Create input field
entry = tk.Entry(window)
entry.pack()

# Create button
button = tk.Button(window, text="Calculate Factorial",
command=calculate_factorial)
button.pack()

# Create result label
result_label = tk.Label(window, text="")
result_label.pack()

# Start the GUI event loop
window.mainloop()
```

Output:



Q.3 Create a Christmas tree using various shapes (Triangle, rectangle(for stem), circle(for decoration), of in Canvas

Code:

```
import tkinter as tk

def draw_christmas_tree(canvas):
    # Draw the tree using larger triangles for a fuller
    # appearance
    canvas.create_polygon(150, 30, 70, 150, 230, 150,
fill="green") # Top triangle
    canvas.create_polygon(150, 100, 50, 250, 250, 250,
fill="green") # Middle triangle
    canvas.create_polygon(150, 180, 0, 350, 300, 350,
fill="green") # Bottom triangle

    # Draw the trunk
    canvas.create_rectangle(135, 350, 165, 400,
fill="saddlebrown") # Tree trunk

    # Draw decorations (circles)
    decorations = [(100, 100), (150, 120), (200, 100), (130,
200), (170, 200)]
    for x, y in decorations:
        canvas.create_oval(x - 8, y - 8, x + 8, y + 8,
fill="red") # Red ornaments

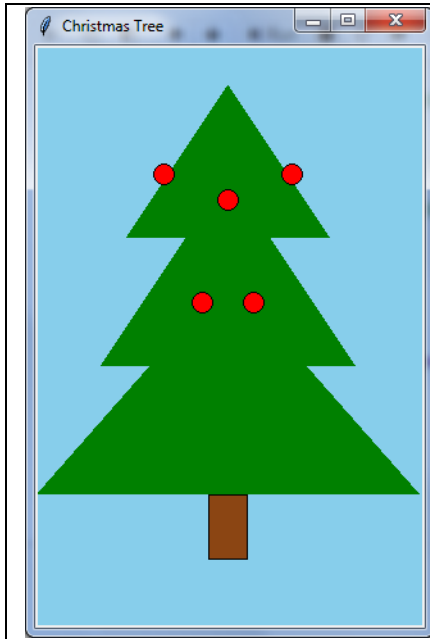
# Create the main window
window = tk.Tk()
window.title("Christmas Tree")

# Create a canvas
canvas = tk.Canvas(window, width=300, height=450,
bg="skyblue")
canvas.pack()

# Draw the Christmas tree
draw_christmas_tree(canvas)

# Start the Tkinter main loop
window.mainloop()
```


Output:



Conclusion: Thus studied GUI Frame and Canvas application using Tkinter

- Chaitanya Shah