

|  |
| --- |
| 8 |
|  |
|  |
|  |

**Experiment No. 8**

           





**Code:**

import tkinter as tk

def greet():

name = entry.get()

gender = gender\_var.get()

greeting\_label.config(text=f"Hello, {name}! You are {gender}.")

# Create the main window

root = tk.Tk()

root.title("Simple GUI")

# Add a Label

label = tk.Label(root, text="Enter your name:")

label.pack()

# Add an Entry widget

entry = tk.Entry(root)

entry.pack()

# Add a Label for gender selection

gender\_label = tk.Label(root, text="Select your gender:")

gender\_label.pack()

# Variable to hold the selected gender

gender\_var = tk.StringVar()

# Add Male and Female Radiobuttons

male\_radio = tk.Radiobutton(root, text="Male", variable=gender\_var, value="male")

male\_radio.pack()

female\_radio = tk.Radiobutton(root, text="Female", variable=gender\_var, value="female")

female\_radio.pack()

# Set default gender

gender\_var.set("male")

# Add a Button

button = tk.Button(root, text="Greet", command=greet)

button.pack()

# Add a Label to display the greeting

greeting\_label = tk.Label(root, text="")

greeting\_label.pack()

# Start the GUI event loop

root.mainloop()

**Output:**

****

**Conclusion:**

Tkinter is the standard GUI (Graphical User Interface) toolkit for Python. It provides a simple and easy-to-use interface for creating GUI applications. Tkinter is based on the Tk GUI toolkit, which originated as a part of the Tcl (Tool Command Language) scripting language. Tkinter comes bundled with Python, so there's no need to install any additional libraries to use it. It provides a set of widgets (UI elements) that can be used to create various types of windows, dialogs, buttons, menus, text boxes, and more.

Widgets are the building blocks of a Tkinter GUI application. Each widget represents a graphical component such as a button, label, entry field, or frame. Tkinter provides three geometry managers to arrange widgets within a window: pack, grid, and place.