

Assignment 5

```
# 1. Write a Tkinter program that displays a window with a
Label widget showing the text "Welcome to
# Tkinter!" in a specific font and color.
```

```
import tkinter as tk
from tkinter import font
```

```
# Create the main window
window = tk.Tk()
window.title("Tkinter Welcome App")
```

```
# Set window size
window.geometry("300x200")
```

```
# Define font and color
custom_font = font.Font(family="Monolisa", size=16, weight
="bold")
text_color = "#3498db" # Hex color code for blue
```

```
# Create a Label widget with custom font and color
label = tk.Label(window, text="Welcome to Tkinter!", font=c
ustom_font, fg=text_color)
```

```
# Position the label in the center of the window
label.pack(pady=50)
```

```
# Run the Tkinter event loop
window.mainloop()
```

```
# 2. Develop a Tkinter application with three Buttons: one
to display a message in a Label, one to
```

```

# change the Label text, and one to exit the application.

import tkinter as tk

# Function to display a message in the label
def display_message():
    label.config(text="Hello, Tkinter!")

# Function to change the label text
def change_label_text():
    label.config(text="You changed the text!")

# Function to exit the application
def exit_application():
    window.destroy()

# Create the main window
window = tk.Tk()
window.title("Tkinter Button Application")

# Set window size
window.geometry("300x200")

# Create a Label widget
label = tk.Label(window, text="", font=("Helvetica", 14))
label.pack(pady=20)

# Create Buttons
display_button = tk.Button(window, text="Display Message",
    command=display_message)
display_button.pack(pady=5)

change_button = tk.Button(window, text="Change Label Text",
    command=change_label_text)
change_button.pack(pady=5)

exit_button = tk.Button(window, text="Exit", command=exit_a

```

```

pplication)
exit_button.pack(pady=5)

# Run the Tkinter event loop
window.mainloop()

# 3. Implement a Tkinter interface with a Text widget for m
ulti-line input and an Entry widget for
# single-line input. Add a Button to copy the content from
the Entry widget into the Text widget.

import tkinter as tk

# Function to copy text from the Entry widget to the Text w
idget
def copy_to_text():
    entry_text = entry.get() # Get text from Entry widget
    text_widget.insert(tk.END, entry_text + "\n") # Insert
text into Text widget with a new line

# Create the main window
window = tk.Tk()
window.title("Text and Entry Widget Example")

# Set window size
window.geometry("400x300")

# Create an Entry widget for single-line input
entry = tk.Entry(window, font=("Helvetica", 14))
entry.pack(pady=10)

# Create a Button to copy the content from Entry to Text
copy_button = tk.Button(window, text="Copy to Text", comman
d=copy_to_text)
copy_button.pack(pady=5)

```

```
# Create a Text widget for multi-line input
text_widget = tk.Text(window, height=10, width=40, font=("H
elvetica", 12))
text_widget.pack(pady=10)
```

```
# Run the Tkinter event loop
window.mainloop()
```

4. Build a Tkinter application with a Frame containing multiple Check Buttons for selecting different options from online clothing store. Include a Button to print the selected options.

```
import tkinter as tk
```

```
# Function to print selected options
```

```
def print_selected_options():
```

```
    selected_options = []
```

```
    for index, var in enumerate(check_vars):
```

```
        if var.get(): # Check if the Checkbutton is select
ed
```

```
            selected_options.append(clothing_options[inde
x])
```

```
    print("Selected options:", selected_options)
```

```
# Create the main window
```

```
window = tk.Tk()
```

```
window.title("Online Clothing Store")
```

```
# Set window size
```

```
window.geometry("300x300")
```

```
# Clothing options
```

```
clothing_options = [
```

```
    "T-Shirt",
```

```
    "Jeans",
```

```
    "Jacket",
```

```

        "Sweater",
        "Shorts",
        "Hat"
    ]

    # Variable list to hold the state of each Checkbutton
    check_vars = []

    # Create a Frame to hold the Checkbuttons
    frame = tk.Frame(window)
    frame.pack(pady=20)

    # Create Checkbuttons for each clothing option
    for option in clothing_options:
        var = tk.IntVar() # Create a variable to hold the state of each Checkbutton
        check_vars.append(var) # Add the variable to the list
        check_button = tk.Checkbutton(frame, text=option, variable=var)
        check_button.pack(anchor='w') # Pack Checkbuttons in a vertical list

    # Create a Button to print selected options
    print_button = tk.Button(window, text="Print Selected Options", command=print_selected_options)
    print_button.pack(pady=10)

    # Run the Tkinter event loop
    window.mainloop()

# 5. Write a Tkinter program with a set of Radio Buttons allowing the user to select the candidate for
# the vote from 3 candidates. Add a Label that updates the count of votes for each candidate when
# button is clicked.
import tkinter as tk

```

```

# Function to update the vote count
def cast_vote():
    selected_candidate = candidate_var.get()
    if selected_candidate:
        votes[selected_candidate] += 1
        update_vote_count()

def update_vote_count():
    count_text = "\n".join([f"{candidate}: {votes[candidate]}" for candidate in candidates])
    vote_count_label.config(text=count_text)

# Create the main window
window = tk.Tk()
window.title("Voting System")

# Set up candidate names and votes
candidates = ["Candidate A", "Candidate B", "Candidate C"]
votes = {candidate: 0 for candidate in candidates}

# Variable to hold the selected candidate
candidate_var = tk.StringVar()

# Create radio buttons for each candidate
for candidate in candidates:
    radio_button = tk.Radiobutton(window, text=candidate, variable=candidate_var, value=candidate)
    radio_button.pack(anchor='w')

# Create a button to cast the vote
vote_button = tk.Button(window, text="Vote", command=cast_vote)
vote_button.pack(pady=10)

# Label to show vote counts
vote_count_label = tk.Label(window, text="")
vote_count_label.pack(pady=10)

```

```

# Run the Tkinter event loop
window.mainloop()

# 6. Develop a Tkinter window with a Text widget that contains enough text to require scrolling. Add a
# Scrollbar that allows the user to scroll through the text.
import tkinter as tk

# Create the main window
window = tk.Tk()
window.title("Text Widget with Scrollbar")

# Create a Text widget
text_widget = tk.Text(window, wrap='word', height=10, width=50)
text_widget.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)

# Create a Scrollbar widget
scrollbar = tk.Scrollbar(window, command=text_widget.yview)
scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

# Configure the Text widget to use the scrollbar
text_widget.config(yscrollcommand=scrollbar.set)

# Insert enough text to require scrolling
for i in range(1, 21):
    text_widget.insert(tk.END, f"This is line number {i}\n")

# Run the Tkinter event loop
window.mainloop()

# 7. Design a Tkinter interface with a Listbox containing a

```

```

list of subjects as items and a Scrollbar to
# navigate through the items in the Listbox.
import tkinter as tk

# Create the main window
window = tk.Tk()
window.title("Listbox with Scrollbar")

# Create a Listbox widget
listbox = tk.Listbox(window, height=10, width=50)
listbox.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)

# Add subjects to the Listbox
subjects = ["Mathematics", "Physics", "Chemistry", "Biolog
y", "History", "Geography", "Computer Science"]
for subject in subjects:
    listbox.insert(tk.END, subject)

# Create a Scrollbar widget
scrollbar = tk.Scrollbar(window, command=listbox.yview)
scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

# Configure the Listbox to use the scrollbar
listbox.config(yscrollcommand=scrollbar.set)

# Run the Tkinter event loop
window.mainloop()

# 8. Write a Tkinter program that includes a Canvas widget
where you draw a line, rectangle, and oval.
# Allow the user to click a Button to change the color of t
he shapes.
import tkinter as tk

# Function to change the color of shapes
def change_color():
    canvas.itemconfig(line, fill="red")

```



```

        canvas.itemconfig(rectangle, fill="green")
        canvas.itemconfig(oval, fill="blue")

# Create the main window
window = tk.Tk()
window.title("Canvas with Shapes")

# Create a Canvas widget
canvas = tk.Canvas(window, width=300, height=200)
canvas.pack()

# Draw shapes
line = canvas.create_line(50, 50, 250, 50, width=2)
rectangle = canvas.create_rectangle(50, 70, 250, 130, fill
="yellow")
oval = canvas.create_oval(50, 140, 250, 180, fill="orange")

# Create a button to change the color of shapes
color_button = tk.Button(window, text="Change Color", comma
nd=change_color)
color_button.pack(pady=10)

# Run the Tkinter event loop
window.mainloop()

# 9. Create a Tkinter window with a Button that, when click
ed, displays a MessageBox with a custom
# message and an OK button. Use multiple buttons to display
instance of each type of Message Box.

import tkinter as tk
from tkinter import messagebox

# Function to show different message boxes
def show_info():
    messagebox.showinfo("Info", "This is an informational m
essage.")

```

```

def show_warning():
    messagebox.showwarning("Warning", "This is a warning message.")

def show_error():
    messagebox.showerror("Error", "This is an error message.")

# Create the main window
window = tk.Tk()
window.title("MessageBox Examples")

# Create buttons to display message boxes
info_button = tk.Button(window, text="Show Info", command=show_info)
info_button.pack(pady=5)

warning_button = tk.Button(window, text="Show Warning", command=show_warning)
warning_button.pack(pady=5)

error_button = tk.Button(window, text="Show Error", command=show_error)
error_button.pack(pady=5)

# Run the Tkinter event loop
window.mainloop()

# 10. Develop a Tkinter application with a Menu at the top containing "File" and "Help" dropdown options. Implement event handling so that clicking "File" opens a sub-menu with "New" and "Open" options, and clicking "Help" shows a MessageBox with help information.
import tkinter as tk

```

```

from tkinter import messagebox

# Function for "New" option
def new_file():
    print("New File created!")

# Function for "Open" option
def open_file():
    print("Open File dialog!")

# Function to show help information
def show_help():
    messagebox.showinfo("Help", "This is the help informati
on.")

# Create the main window
window = tk.Tk()
window.title("Menu Example")

# Create a Menu
menu = tk.Menu(window)

# Create a "File" menu
file_menu = tk.Menu(menu, tearoff=0)
file_menu.add_command(label="New", command=new_file)
file_menu.add_command(label="Open", command=open_file)
menu.add_cascade(label="File", menu=file_menu)

# Create a "Help" menu
help_menu = tk.Menu(menu, tearoff=0)
help_menu.add_command(label="Help", command=show_help)
menu.add_cascade(label="Help", menu=help_menu)

# Configure the window to use the menu
window.config(menu=menu)

# Run the Tkinter event loop
window.mainloop()

```