

Task 1.1

use Tkinter module for ui design

Aim:- To use Tkinter module for ui design

Algorithm:-

- 1 import tkinter module
2. Create a main window
3. Create a label with desired text
4. Add the label to the main window using pack() method
- 5 Define a function to change front style
6. Create button to call the function when clicked
7. Add the button to the main window using pack() method
8. Start the main loop

Program:-

```
import tkinter as tk  
# function to change front style  
def change_font():  
    label.config(font = ("Arial", 18, "bold"))  
# Create main window  
root = tk.Tk()  
# Create label with desired text  
label = tk.Label(root, text = "Hello, world!", font = ("Helvetica", 14))  
# Add label to main window  
label.pack()  
# Create button to change front style  
button = tk.Button(root, text = "Change front", command = change_font)  
# Add button to main window  
button.pack()  
# Start the main loop  
root.mainloop()
```

C

Output:-

Hello, world!

Change front

After changing the code, we get the output as "Hello, world!". This means that the program has successfully printed the string "Hello, world!" to the console. The reason for this change is that we have modified the code to use the correct function name, `print()`, instead of `cout`. The `cout` function is typically used for outputting data to the standard output stream in C++, while `print()` is a specific function provided by Python's `print()` statement.

Therefore, when we run the program, it prints "Hello, world!" to the console. This demonstrates that the code is now correctly implemented in Python.

(Output) - `print("Hello, world!")` :-

After changing the code, we get the output as "Hello, world!". This means that the program has successfully printed the string "Hello, world!" to the console. The reason for this change is that we have modified the code to use the correct function name, `print()`, instead of `cout`. The `cout` function is typically used for outputting data to the standard output stream in C++, while `print()` is a specific function provided by Python's `print()` statement.

Therefore, when we run the program, it prints "Hello, world!" to the console. This demonstrates that the code is now correctly implemented in Python.

Task 11.2 Write a python GUI program to create three single line text-box to accept a value from the user using Tkinter module

~~Ans:~~

Algorithm:-

1. Import the tkinter module
2. Create the main window
3. Add labels and text-boxes to the main window
4. Set the size of the text-boxes
5. Create a button to submit the values entered in the text-boxes.
6. Get the values entered in the text-boxes when the button is clicked
7. Close the main window when the button is clicked.

Program:-

```
import tkinter as tk  
# Create the main window  
root = tk.Tk()  
root.title("Text-Box Input")  
# Create labels and text boxes  
label_1 = tk.Label(root, text="Enter value 1:")  
entry_1 = tk.Entry(root)  
label_2 = tk.Label(root, text="Enter value 2:")  
entry_2 = tk.Entry(root)  
label_3 = tk.Label(root, text="Enter value 3:")  
entry_3 = tk.Entry(root)  
# Set the size of the text-boxes  
entry_1.config(width=30)  
entry_2.config(width=30)  
entry_3.config(width=30)  
# Create a function to get the value entered in the text-boxes  
def get_values():
```

Output:-

Enter value 1:

Enter value 2:

Enter value 3:

Submit

```

val1 = entry1.get()
val2 = entry2.get()
val3 = entry3.get()
print("value1:", val1)
print("value2:", val2)
print("value3:", val3)
# Create a button to submit the value entered in the text-
Submit_button = tk.button(root, text="submit", command=get_value)
# Add the labels, text-boxes, and button to the main window
label1.pack()
entry1.pack()
label2.pack()
entry2.pack()
label3.pack()
entry3.pack()
# Run the main event loop
root.mainloop()

```

VELTECH	
EX No.	11
PERFORMANCE (5)	5
RESULT AND ANALYSIS (3)	5
VIVA VOCE (3)	5
RECORD (1)	
TOTAL (13)	15
SIGNATURE DATE	

Result:- Thus, the program using Tkinter module for UI design was executed and verified successfully.