Date:29.10.25

TASK:12

Implementation of Computer Interaction

PROBLEM STATEMENT :

Design and implement a simple interactive software application that demonstrates the principles of Human-Computer Interaction (HCI), ensuring that the user interface is intuitive, responsive, and accessible to users with minimal training.

AIM:

To implement a user-centered interface using HCI principles that allows seamless interaction between humans and computers through an easy-to-use GUI.

OBJECTIVE:

1. understand the principles of Human-Computer Interaction.
2.To design an interactive GUI that is user-friendly and visually accessible.
3. To implement basic interaction using standard GUI components.
4. To apply usability principles like feedback, consistency, and simplicity.
5. To test the GUI for responsiveness and error handling.

DESCRIPTION :

Human-Computer Interaction (HCI) focuses on designing systems that people find easy to learn and effective to use. In this project, we demonstrate HCI implementation by building a basic GUI-based calculator in Python using Tkinter a standard GUI library.

ALGORITHM:

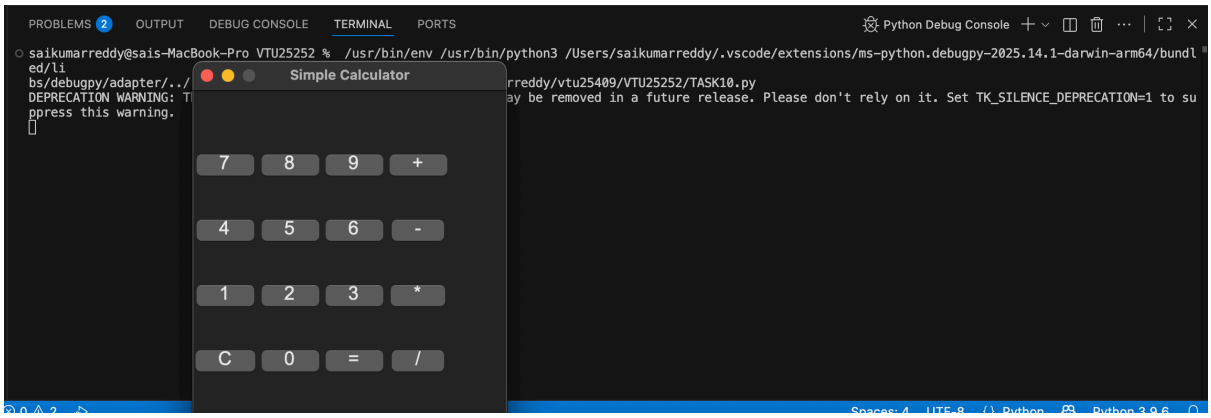
Step 1:Start
Step 2: Design the GUI layout with buttons (0–9, +, –, ×, ÷, =) and display area.
Step 3: Capture user input through button clicks.
Step 4:Display input in a text field as users press buttons.
Step 5: On pressing '=', evaluate the expression and show the result.
Step 6: If the expression is invalid, show an error message.
Step 7: Provide a clear button to reset the interface.
Step 8: End

PROGRAM :

```
import tkinter as tk
from tkinter import messagebox

def press(num):
    entry_var.set(entry_var.get() + str(num))

def clear():
    entry_var.set("")

def equal():
    try:
        result = str(eval(entry_var.get()))
        entry_var.set(result)
    except Exception as e:
        messagebox.showerror("Error", "Invalid Input")

# Create main window
root = tk.Tk()
root.title("Simple Calculator")
root.geometry("300x400")
root.resizable(False, False)

entry_var = tk.StringVar()

# Entry widget for input/output
entry = tk.Entry(root, textvariable=entry_var, font=('Arial', 20), bd=10,
insertwidth=2, width=14,
                 borderwidth=4, justify='right')
entry.grid(row=0, column=0, columnspan=4)

# Button layout
buttons = [
    ('7', 1, 0), ('8', 1, 1), ('9', 1, 2), ('+', 1, 3),
    ('4', 2, 0), ('5', 2, 1), ('6', 2, 2), ('-', 2, 3),
    ('1', 3, 0), ('2', 3, 1), ('3', 3, 2), ('*', 3, 3),
    ('C', 4, 0), ('0', 4, 1), ('=', 4, 2), ('/', 4, 3)
]

for (text, row, col) in buttons:
    if text == '=':
        tk.Button(root, text=text, padx=20, pady=20, bd=8, font=('Arial', 18),
```

```
            command=equal).grid(row=row, column=col)
    elif text == 'C':
        tk.Button(root, text=text, padx=20, pady=20, bd=8, font=('Arial', 18),
            command=clear).grid(row=row, column=col)
    else:
        tk.Button(root, text=text, padx=20, pady=20, bd=8, font=('Arial', 18),
            command=lambda t=text: press(t)).grid(row=row, column=col)

root.mainloop()
```

OUTPUT :



Result: Thus to implement a user-centered interface using HCI principles that allows seamless interaction between humans and computers through an easy-to-use GUI has been implemented successfully.