Appendix 1

**MINISTRY FOR DEVELOPMENT OF INFORMATION TECHNOLOGIES AND COMMUNICATIONS OF THE REPUBLIC OF UZBEKISTAN**

**TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES**

**NAMED AFTER MUHAMMAD AL-KHORAZMI**

Department of "Power Supply Systems"

**Practical task #1**

***By subject***

***“Software Systems Design”***

Completed by G`ulomnazirov Muslimbek

Full name of the student.

Group 319-22

Received by: Mo'tabar Abduxakimova

**Tashkent 2024**

1-method to create calculator  
Topic : creating calculator app in PYTHON with Flet ,   
  
 Theoretical aspects of FTET  
Flet is a Python framework designed for building interactive web, desktop, and mobile applications using Python, similar to how you would develop apps using Flutter (Google’s UI toolkit). Flet simplifies the development process by eliminating the need to deal with HTML, CSS, or JavaScript. Instead, you define your app’s UI and logic using pure Python code, and Flet takes care of rendering the UI across different platforms.

Key Features of Flet:

1. Cross-Platform: Flet allows you to create apps for web, desktop (Windows, macOS, Linux), and mobile (Android, iOS) using one codebase.

2. Flutter-Powered: Flet is built on top of Flutter, meaning it leverages Flutter’s reactive UI model but without requiring developers to write Dart code.

3. Interactive UI: With Flet, you can create highly interactive user interfaces (buttons, text fields, sliders, etc.) just using Python.

4. Real-Time Updates: It allows real-time updates to the app’s UI without the need for manual page reloads.

5. Server-Side Execution: Your Python code runs on the server side, meaning no Python code is executed on the client side (browser or mobile). This is different from typical web frameworks like Flask or Django.

How Flet Works:

• Declarative UI: Similar to frameworks like React, Flet uses a declarative programming model where the UI is described as a composition of components (buttons, text fields, etc.).

• Event Handling: You can handle user events (like button clicks, form submissions) using Python functions, making it intuitive for Python developers.

Example of Flet Usage:  
  
import flet as ft

def main(page: ft.Page):

# Define UI components

text = ft.Text("Hello, Flet!")

button = ft.ElevatedButton(text="Click Me!", on\_click=lambda e: print("Button clicked"))

# Add components to the page

page.add(text, button)

# Run the app  
ft.app(target=main)

**Alternatives to Flet in Python:**

1. **Tkinter**:

• **Overview**: A built-in Python library for creating desktop GUIs.

• **Advantages**:

• Comes pre-installed with Python, making it easy to start without extra dependencies.

• Suitable for simple desktop applications.

• **Disadvantages**:

• Limited in terms of modern design and cross-platform support.

• No support for mobile or web apps.

2. **Kivy**:

• **Overview**: A Python framework for developing multi-touch applications. It supports Windows, Linux, OS X, Android, and iOS.

• **Advantages**:

• Excellent for multi-touch devices and mobile apps.

• More flexible in terms of creating custom UI.

• **Disadvantages**:

• Can be harder to use compared to Flet due to its complexity.

• No built-in support for web apps.

3. **PyQt/PySide**:

• **Overview**: Python bindings for the Qt toolkit, allowing you to create highly sophisticated desktop applications.

• **Advantages**:

• Very powerful and feature-rich for creating professional desktop applications.

• Excellent support for custom widgets and layouts.

• **Disadvantages**:

• Heavier and more complex to learn than Flet or Tkinter.

• Primarily focused on desktop applications, no web or mobile support out of the box.

4. **Flask/Django** (For web development):

• **Overview**: Flask is a lightweight web framework, and Django is a full-stack web framework.

• **Advantages**:

• Suitable for building large-scale web applications.

• Django offers an all-in-one solution, including ORM, templating, and admin panels.

• **Disadvantages**:

• Flask and Django require a deeper understanding of web technologies (HTML, CSS, JavaScript).

• They don’t natively support mobile or desktop app development.

**Advantages of Flet Over Alternatives:**

1. **Unified Development**:

• With Flet, you can build for web, desktop, and mobile with one codebase. Alternatives like Tkinter and PyQt focus primarily on desktop applications, while Flask and Django are limited to web development.

2. **Simplified UI Code**:

• Flet abstracts away the need to write HTML, CSS, or JavaScript for building web UIs. Compared to Flask/Django, where you’d need to handle front-end code separately, Flet lets you build the front-end with just Python.

3. **Cross-Platform Support**:

• Unlike Tkinter or PyQt, which focus on desktop GUIs, Flet allows you to deploy the same app on web, desktop, and mobile platforms without additional code changes.

4. **Ease of Use**:

• Flet provides a higher level of abstraction, making it simpler to start building user interfaces quickly compared to more complex frameworks like PyQt or Kivy.

**Python Keywords and Key Aspects**

**Python Keywords:**

Keywords are reserved words in Python that have a specific meaning and are used to define the structure and syntax of the code. You cannot use these keywords as variable names, function names, or any other identifiers.

**Common Python Keywords**:

1. if**,** else**,** elif: Control flow keywords for conditional statements.

2. for**,** while: Looping constructs.

3. def: Used to define functions.

4. class: Defines a class.

5. return: Exits a function and returns a value.

6. try**,** except: Keywords used for handling exceptions.

7. import: Used to import modules.

8. with: Simplifies exception handling (context management).

**Key Aspects of Python:**

1. **Dynamic Typing**: Python variables do not need explicit type declarations. The type is determined at runtime.

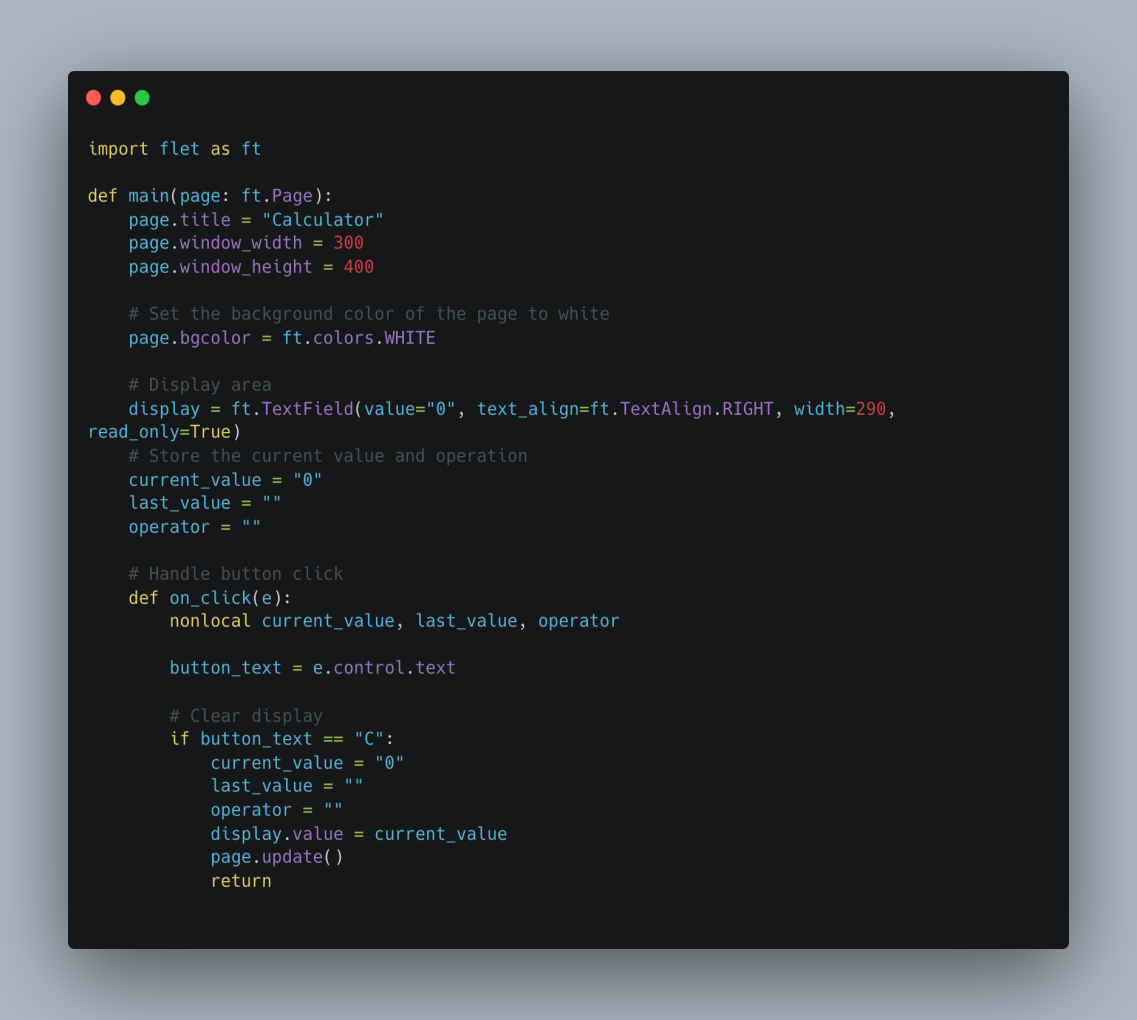
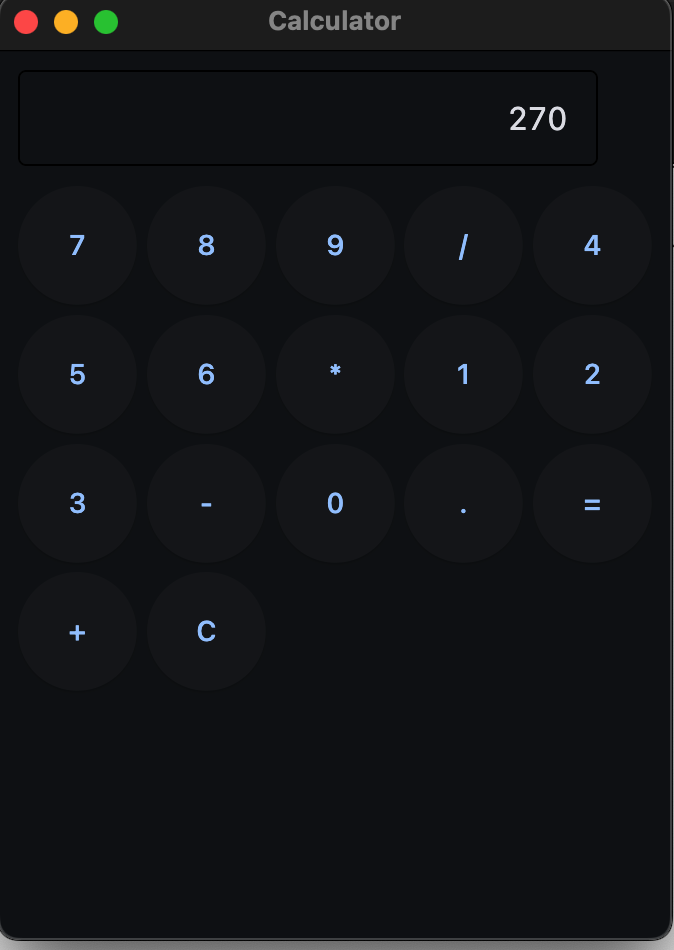
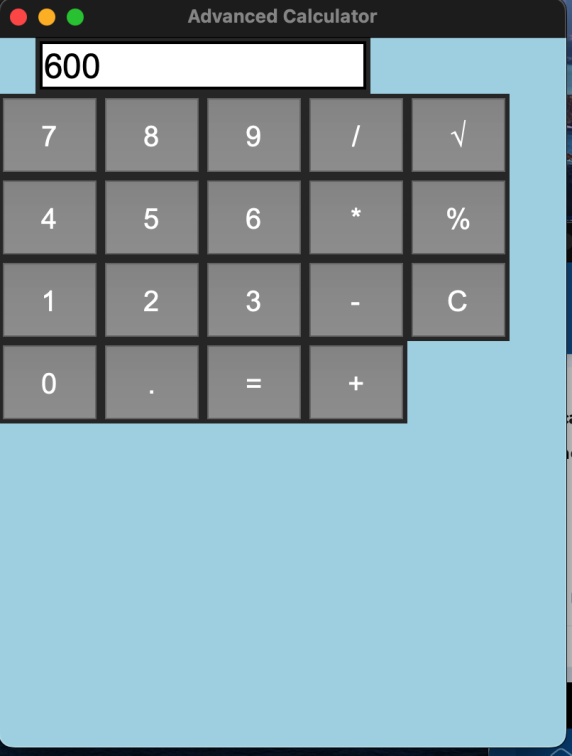
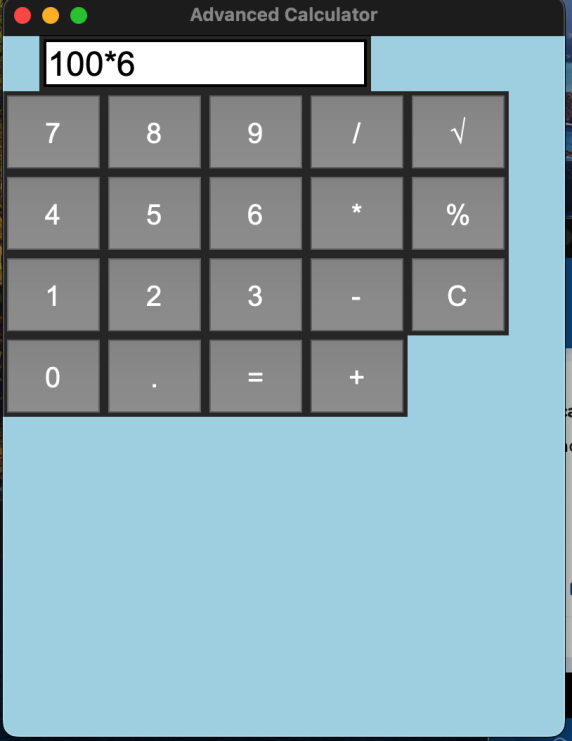
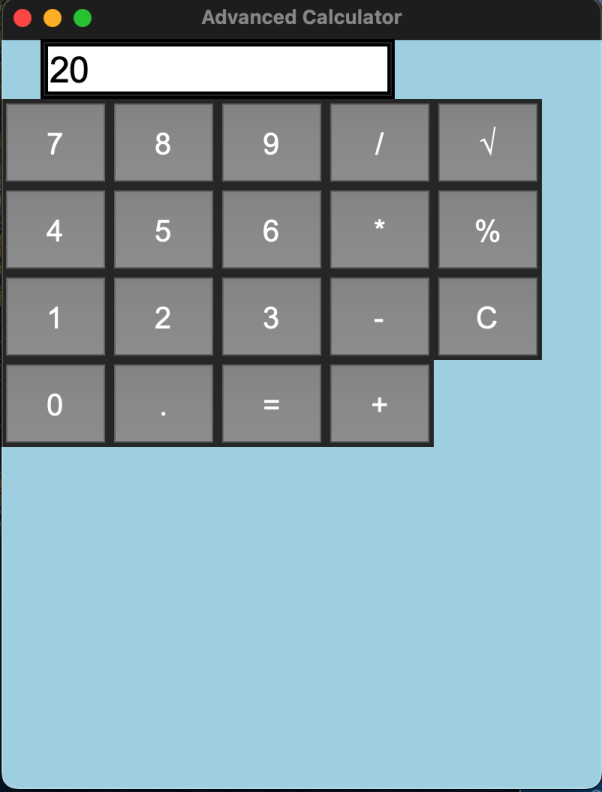
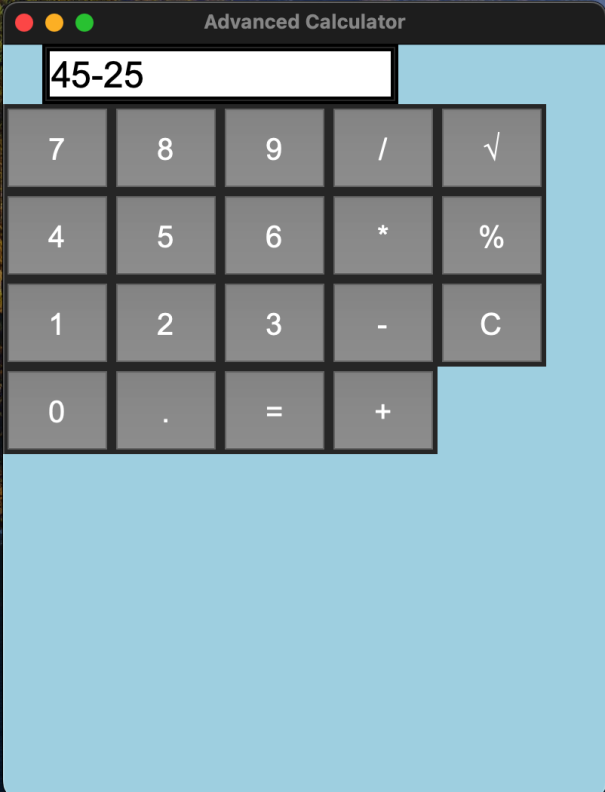
2. **Interpreted Language**: Python code is executed line by line, which makes it easy to debug but can be slower compared to compiled languages.

3. **Indentation**: Python uses indentation to define code blocks instead of braces {} like many other languages.

4. **Object-Oriented**: Python supports classes and objects, promoting code reuse and modular design.

5. **Standard Library**: Python comes with an extensive standard library, providing modules for handling various tasks such as file I/O, networking, and regular expressions.

6. **Extensibility**: Python can be extended with modules written in C or C++ to improve performance.

Practical Part  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
2-method to create calculator.  
  
Topic : creating calculator app in PYTHON with Tkinter ,   
  
  
  
  
   
  
  
  
  
  
Calculator   
  
  
  
  
  
 In conclusion   
  
Flet is a great tool for Python developers who want to build cross-platform applications without worrying about web technologies like HTML or CSS.

Compared to other frameworks like Tkinter, PyQt, Kivy, or Django, Flet provides simplicity, cross-platform capabilities, and easier development for web and mobile.  
Python Keywords are essential elements that define Python’s syntax and behavior, while its key aspects like dynamic typing, interpreted nature, and object-oriented features make it versatile and beginner-friendly.