**Simple phonebook application in Python**

**Introduction:**

A simple phonebook application in Python can be a great way to practice fundamental programming concepts like functions, loops, conditionals, and error handling.

**Step-by-Step Implementation**

1**. Setup Your Project**

Create a new Python file named `phonebook.py`.

**2. Initialize the Phonebook**

At the program's start, an empty dictionary stores contact information. Each name is the key, and the phone number is the value.

**3. User Interface**

A loop continuously shows a menu to the user until they choose to exit.

**4. Implement CRUD Operations**

You can add, search for, delete, and list all contacts.

**Source Code**

**Step1:** python

# phonebook.py

**Step 2:** Initialize the phonebook as an empty dictionary

phonebook = {}

**Step 3:** Define a function to display the menu

def display\_menu():

print("\nPhonebook Menu:")

print("1. Add New Contact")

print("2. Search for a Contact")

print("3. Delete a Contact")

print("4. List All Contacts")

print("5. Exit")

**Step 4:** Implementing CRUD Operations

# Add a new contact

def add\_contact():

name = input("Enter the contact's name: ").capitalize()

if name in phonebook:

print(f"{name} already exists in the phonebook.")

else:

phone = input(f"Enter {name}'s phone number: ")

phonebook[name] = phone

print(f"{name} added successfully!")

# Search for a contact

def search\_contact():

name = input("Enter the name to search: ").capitalize()

if name in phonebook:

print(f"{name}'s phone number is {phonebook[name]}")

else:

print(f"{name} not found in the phonebook.")

# Delete a contact

def delete\_contact():

name = input("Enter the name to delete: ").capitalize()

if name in phonebook:

del phonebook[name]

print(f"{name} has been deleted.")

else:

print(f"{name} not found in the phonebook.")

# List all contacts

def list\_contacts():

if phonebook:

print("\nPhonebook Entries:")

for name, phone in phonebook.items():

print(f"Name: {name}, Phone: {phone}")

else:

print("The phonebook is empty.")

# Main function to handle the menu and user actions

def main():

while True:

display\_menu()

choice = input("Enter your choice (1-5): ")

if choice == '1':

add\_contact()

elif choice == '2':

search\_contact()

elif choice == '3':

delete\_contact()

elif choice == '4':

list\_contacts()

elif choice == '5':

print("Exiting Phonebook. Goodbye!")

break

else:

print("Invalid choice. Please enter a number between 1 and 5.")

**Step 5:** Run the application

if \_\_name\_\_ == "\_\_main\_\_":

main()

```

**Explanation of the Code**

**- phonebook = {}:** Initializes an empty dictionary to store the contact information.

**- display\_menu():** Displays the menu options to the user.

- **add\_contact():** Prompts the user for a name and phone number, checks if the name already exists, and adds the contact if it doesn’t.

**- search\_contact():** Looks up a contact by name and displays the associated phone number.

- **delete\_contact():** Deletes a contact by name if it exists in the phonebook.

- **list\_contacts():** Lists all contacts with their names and phone numbers.

**- main():** This is the main loop that keeps running until the user selects the exit option. It calls the appropriate functions based on the user's input.

**Testing Scenarios**

**1. Add a New Contact:** Test by adding a new contact to ensure it gets stored correctly.

**2. Search for a Contact:** Verify that searching for an existing contact retrieves the correct phone number.

**3. Delete a Contact:** Check that deleting a contact removes it from the phonebook.

**4. List All Contacts:** Ensure all contacts are listed with their details.

**5. Exit:** Confirm that selecting exit closes the application.

**Example Interaction**

Phonebook Menu:

1. Add New Contact

2. Search for a Contact

3. Delete a Contact

4. List All Contacts

5. Exit

Enter your choice (1-5): 1

Enter the contact's name: John

Enter John's phone number: 1234567890

John added successfully!

Phonebook Menu:

1. Add New Contact

2. Search for a Contact

3. Delete a Contact

4. List All Contacts

5. Exit

Enter your choice (1-5): 2

Enter the name to search: John

John's phone number is 1234567890

Phonebook Menu:

1. Add New Contact

2. Search for a Contact

3. Delete a Contact

4. List All Contacts

5. Exit

Enter your choice (1-5): 4

Phonebook Entries:

Name: John, Phone: 1234567890

**Conclusion**

This Python-based Phonebook application demonstrates CRUD operations with dictionaries and user interaction using loops and conditionals.