



الجمهورية العربية السورية

كلية الهمك

هندسة الاتصالات والالكترونيات

السنة الخامسة

وظيفة شبكات 2

إعداد الطلاب :

لجين حكمت الشمالي

ديما بشار الصارم

إشراف :

د. مهند عيسى

Question 1: Bank ATM Application with TCP Server/Client and Multi-threading

Project Description:

Build a TCP server and client Bank ATM application using Python. The server should handle multiple client connections simultaneously using multi-threading. The application should allow clients to connect, perform banking operations (such as check balance, deposit, and withdraw), and receive their updated account status upon completion.

كود server :

```
server.py - C:\Users\ALAMEEN\Desktop\homework\server.py (3.11.3)
File Edit Format Run Options Window Help

import socket
import threading

# Bank account details
accounts = {
    '1111': 100000,
    '2222': 200000,
    '3333': 300000,
    '4444': 400000
}

def handle_client(client_socket):
    account_number = client_socket.recv(1024).decode()
    if account_number in accounts:
        client_socket.send(b"Welcome! You have connected to the bank server.")
    else:
        client_socket.send(b"Invalid account number. Connection terminated.")
        client_socket.close()
        return

    while True:
        option = client_socket.recv(1024).decode()

        if option == '1':
            balance = accounts[account_number]
            client_socket.send(f"Your current balance is: {balance}".encode())
        elif option == '2':
            amount = int(client_socket.recv(1024).decode())
            accounts[account_number] += amount
            client_socket.send(f"Deposit successful. Your new balance is: {accounts[account_number]}".encode())
        elif option == '3':
            amount = int(client_socket.recv(1024).decode())
            if accounts[account_number] >= amount:
                accounts[account_number] -= amount
                client_socket.send(f"Withdrawal successful. Your new balance is: {accounts[account_number]}".encode())
            else:
                client_socket.send("Insufficient funds. Withdrawal failed.".encode())
        else:
            break
```

كود client :

client.py - C:\Users\ALAMEEN\Desktop\homework\client.py (3.11.3)

File Edit Format Run Options Window Help

```
import socket

def main():
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect(('127.0.0.1', 12345))

    account_number = input("Enter your account number: ")
    client.send(account_number.encode())

    print(client.recv(1024).decode())

    while True:
        print("\nOptions:")
        print("1. Check Balance")
        print("2. Deposit")
        print("3. Withdraw")
        print("4. Exit")
        option = input("Enter option: ")

        if option == '4':
            break

        client.send(option.encode())

        if option == '1':
            print(client.recv(1024).decode())
        elif option == '2' or option == '3':
            amount = input("Enter amount: ")
            client.send(amount.encode())
            print(client.recv(1024).decode())

    client.close()

if __name__ == "__main__":
    main()
```

الخروج :

```
Enter your account number: 1111
Welcome! You have connected to the bank server.

Options:
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter option: 1
Your current balance is: 100000


Options:
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter option: 2
Enter amount: 100000
Deposit successful. Your new balance is: 200000

Options:
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter option: 3
Enter amount: 100000
Withdrawal successful. Your new balance is: 100000

Options:
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter option: 4
```

Question 2: Simple Website Project with Python Flask Framework (you have choice to use Django or any Other Deferent Useful Python Project “from provide Project Links”)

Create a simple website with multiple pages using Flask, HTML, CSS, and Bootstrap. The website should demonstrate your understanding of web design principles.

 Q2.py - C:\Users\ALAMEEN\Desktop\Q2.py (3.11.3)

File Edit Format Run Options Window Help

```
from tkinter import *
# Creating a GUI Window
window = Tk()
def from_kg():
    gram = float(e2_value.get())*1000
    pound = float(e2_value.get())*2.20462
    ounce = float(e2_value.get())*35.274
    t1.delete("1.0",END)
    t1.insert(END, gram)
    t2.delete("1.0", END)
    t2.insert(END, pound)
    t3.delete("1.0", END)
    t3.insert(END, ounce)

e1 = Label(window, text="Input the weight in KG")
e2_value = StringVar()
e2 = Entry(window, textvariable=e2_value)
e3 = Label(window, text="Gram")
e4 = Label(window, text="Pound")
e5 = Label(window, text="Ounce")

t1 = Text(window, height=5, width=30)
t2 = Text(window, height=5, width=30)
t3 = Text(window, height=5, width=30)

b1 = Button(window, text="Convert", command=from_kg)

e1.grid(row=0, column=0)
e2.grid(row=0, column=1)
e3.grid(row=1, column=0)
e4.grid(row=1, column=1)
e5.grid(row=1, column=2)
t1.grid(row=2, column=0)
t2.grid(row=2, column=1)
t3.grid(row=2, column=2)
b1.grid(row=0, column=2)

window.mainloop()
```

الخرج :

Gram	Pound	Ounce
50000.0	110.231	1763.7

Weight Converter GUI with Python

شرح الكود :

هذا الكود مكتوب بلغة Python ويستخدم مكتبة Tkinter لإنشاء واجهة مستخدم رسومية بسيطة لتحويل الوزن بين وحدات الكيلوجرام (كجم)، جرام (جم)، واونصة (أوقية).

استيراد المكتبة:

• `from tkinter import *` : يتم استيراد جميع عناصر واجهة المستخدم الرسومية من مكتبة Tkinter.

إنشاء النافذة الرئيسية:

• `window = Tk()` : إنشاء نافذة رئيسية باسم `window` باستخدام Tkinter.

تعريف دالة التحويل: `(from_kg)`

• `def from_kg():` : يتم تعريف دالة `from_kg` لمسؤولية التحويل.
• خطوات الدالة:
1. قراءة قيمة الإدخال:
 ▪ `gram = float(e2_value.get()) * 1000:`

استخراج قيمة الإدخال المخزنة في المتغير `e2_value` وتحويلها إلى رقم عشري، ثم ضربها بـ 1000 لتحويلها إلى جرام.

```
pound = float(e2_value.get()) * 2.20462:
تحويل قيمة الإدخال إلى Pound (اونصة) باستخدام معامل التحويل 2.20462.
```

```
ounce = float(e2_value.get()) * 35.274:
تحويل قيمة الإدخال إلى Ounce (أوقية) باستخدام معامل التحويل 35.274.
```

2. مسح وتحديث حقول العرض:

```
t1.delete("1.0", END):
حقل عرض جرام t1.
t1.insert(END, gram):
المحسوبة في حقل عرض جرام t1.
(يتم تكرار نفس الأمر لحقل عرض Pound و Ounce)
```

إنشاء عناصر واجهة المستخدم:

- يتم إنشاء عناصر واجهة المستخدم المختلفة باستخدام فئات من مكتبة Tkinter:

- Label: لإنشاء نصوص توضيحية (e.g., Input the weight in KG).
- Entry: لحقل إدخال قيمة الوزن. (e2).
- Text: لعرض نتائج التحويل. (t1, t2, t3).
- Button: لزر التشغيل الذي يستدعي دالة التحويل. (b1).

تخطيط عناصر واجهة المستخدم:

- يتم استخدام طريقة grid لترتيب عناصر واجهة المستخدم على النافذة الرئيسية. window.
- يتم تحديد الصفوف والأعمدة لكل عنصر.

تشغيل البرنامج:

- `window.mainloop()`: يتم تشغيل حلقة Tkinter الرئيسية، مما يسمح للمستخدم بالتفاعل مع واجهة المستخدم الرسومية.