Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and electrical engineering

5th , Network Programming : Homework No1

الجمهورية العربية السورية

اللاذقية -جامعــة تشريـــن

كلية الهندسة الكهربائية والميكانيكية قسم هندسة الاتصالات والالكترونيات

السنة الخامسة: وظيفة1 برمجة شبكات

Name: Siba Ahmad Mustafa, Number:2724 , Submitted To GitHub: https://github.com/SibaMustafa

First Network Programming Homework

Question 1: Python Basics?

A-

If you have two lists, L1=[‘HTTP’,’HTTPS’,’FTP’,’DNS’] L2=[80,443,21,53], convert it to generate this
dictionary **d={**‘HTTP’:80,’HTTPS’:443,’FTP’:21,’DNS’:53 **}**

```
L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
L2 = [80, 443, 21, 53]

d = dict(zip(L1, L2))
print(d)
```

```
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}

Process finished with exit code 0
```

الكود يقوم بإنشاء قاموس (dictionary) في لغة البايثون باستخدام قائمتين (lists) في هذا السياق،
القائمة L1 تحتوي على أسماء بعض البروتوكولات مثل'HTTP', 'HTTPS', 'FTP', 'DNS' ،
والقائمة L2 تحتوي على أرقام تعريف البورت (port numbers) المقابلة لكل بروتوكول على التوالي.

عند استخدام الدالة zip() ، يتم دمج العناصر المتناظرة من القائمتين L1 و L2 معًا لإنشاء أزواج (tuples) ثم يتم تحويل هذه الأزواج إلى قاموس باستخدام الدالة dict()

أخيرًا، يتم طباعة القاموس الناتج d الذي يحتوي على البيانات التالية:
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
هذا يعني أن كل بروتوكول في L1 يتم تعيينه برقم تعريف البورت المقابل له في L2 في القاموس d.

أي أن هذا الكود يقوم بإنشاء قاموس (dictionary) في لغة البايثون يربط بين أسماء بعض البروتوكولات مثل 'HTTP', 'HTTPS', 'FTP', 'DNS" وأرقام تعريف البورت (port numbers) المقابلة لها ٨٠، ٤٤٣، ٢١، ٥٣ على التوالي

B-

Write a Python program that calculates the factorial of a given number entered by user.

```python
def fac(n):
    if n == 0:
        return 1
    else:
        return n * fac(n - 1)
num = int(input("Enter a number: "))
if num < 0:
    print("Factorial is not defined for negative numbers.")
else:
    result = fac(num)
    print(f"The factorial of {num} is {result}")

```

EXAMPLE:

```
Enter a number: 4
The factorial of 4 is 24
```

هذا الكود يحتوي على دالة باسم fac() تقوم بحساب عاملي العدد المحدد. إليك شرح خطوات الكود:

١ .تعريف الدالة:fac(n)

- إذا كان العدد n يساوي صفر(n == 0) ، فإن الدالة تقوم بإرجاع قيمة ١ لأن العاملي للصفر يساوي ١.
- في حالة أخرى، الدالة تقوم بإرجاع العدد n مضروبًا بناتج استدعاء الدالة نفسها بوضع قيمة (n-1) كوسيط.

٢ .يتم طلب إدخال عدد من المستخدم باستخدام دالة input() وتحويل القيمة المدخلة إلى عدد صحيح باستخدام.int()

٣ .يتم فحص إذا كان العدد الذي أدخله المستخدم أقل من صفر، في حال كان كذلك يتم طباعة رسالة تفيد بأن العاملي غير معرف للأعداد السالبة.

٤ .في حالة عدم تواجد أعداد سالبة، يتم حساب العاملي للعدد الذي أدخله المستخدم باستدعاء الدالة fac() وتخزين الناتج في متغير.result

٥. يتم طباعة رسالة تظهر قيمة العدد المدخل وناتج العاملي الخاص به

باختصار، هذا الكود يسمح للمستخدم بإدخال عدد صحيح ويقوم بحساب عاملي هذا العدد إذا كان العدد غير سالب، وإذا كان العدد سالبًا يتم طباعة رسالة تفيد بأن عاملي العدد غير معرف.

C-

L=[‘Network’ , ’Bio’ , ’Programming’, ‘Physics’ , ‘Music’]

In this exercise, you will implement a Python program that reads the items of the previous list and identifies the items that starts with ‘B’ letter, then print it on screen.

```
L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']

items_b = [x for x in L if x.startswith('B')]

if items_b:
    print("Items starting with 'B':")
    for x in items_b:
        print(x)
else:
    print("No items found starting with 'B'")
```

```
Items starting with 'B':
Bio
```

هذا الكود يقوم بالبحث في قائمة L عن العناصر التي تبدأ بحرف'B' ، وإذا تم العثور على عناصر كهذه، يتم طباعتها

١ .يتم تعريف قائمة L تحتوي على العناصر التالية.['Network', 'Bio', 'Programming', 'Physics', 'Music'] :

٢ .يستخدم التعبير التالي لإنشاء قائمة جديدة items_b تحتوي على العناصر من قائمة L التي تبدأ بحرف:'B'

```
items_b = [x for x in L if x.startswith('B')]
```

٣ .يتم التحقق مما إذا كانت القائمة items_b غير فارغة باستخدام(if items_b) ، وإذا كانت غير فارغة يتم طباعة رسالة "Items starting with 'B':".

٤ .في حالة وجود عناصر تبدأ بحرف'B' ، يتم طباعة كل عنصر بشكل منفصل باستخدام حلقة:for

```
for x in items_b:
    print(x)
```

٥ .إذا لم يتم العثور على عناصر تبدأ بحرف'B' ، يتم طباعة رسالة تفيد بأنه لم يتم العثور على عناصر تبدأ بحرف 'B' باستخدام else:

```
print("No items found starting with 'B'")
```

أي أن الكود يقوم بالبحث في قائمة العناصر L عن العناصر التي تبدأ بحرف 'B'، وإذا تم العثور على عناصر كهذه يتم طباعتها، وإذا لم يتم العثور على أي عناصر يتم طباعة رسالة بأنه لم يتم العثور على عناصر تبدأ بحرف 'B'.

D-Using Dictionary comprehension, Generate this dictionary
d={0:1,1:2,2:3,3:4,4:5,5:6,6:7,7:8,8:9,9:10,10:11}

```
1  d = {i: i+1 for i in range(11)}
2  print(d)
```

```
{0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}

Process finished with exit code 0
```

الكود يستخدم تعبير قائمة القاموس (dictionary comprehension)

في لغة بايثون يتم استخدام التعبير {i: i+1 for i in range(11)} لإنشاء قاموس يحتوي على مفاتيح تتكون من الأرقام من ٠ إلى ١٠، وقيم كل مفتاح تكون الرقم السابق له بواحد.

بمعنى آخر، يتم توليد القاموس بحيث يكون لكل مفتاح قيمة تكون تساوي المفتاح + ١، وذلك باستخدام التعبير i+1 في الجزء الأيمن من التعبير. بعد ذلك، يتم طباعة القاموس الناتج باستخدام`print(d)` ، حيث تظهر جميع الأزواج المفتاح-القيمة في القاموس.

باختصار، الكود يقوم بإنشاء قاموس يحتوي على أزواج المفتاح-القيمة حيث تكون القيمة تساوي المفتاح + ١، ثم يتم طباعة هذا القاموس.

Question 2: Convert from Binary to Decimal

Write a Python program that converts a Binary number into its equivalent Decimal number.
The program should start reading the binary number from the user. Then the decimal equivalent number must be calculated. Finally, the program must display the equivalent decimal number on the screen.

```python
def binary_to_decimal(binary):
    decimal = 0
    for digit in binary:
        if digit != '0' and digit != '1':
            return None
        decimal = decimal * 2 + int(digit)
    return decimal
binary_num = input("Enter a binary number: ")
decimal_num = binary_to_decimal(binary_num)

if decimal_num is not None:
    print(f"The decimal equivalent of {binary_num} is: {decimal_num}")
else:
    print("Invalid input.")
```

```
Enter a binary number: 10101011
The decimal equivalent of 10101011 is: 171
```

هذا الكود في Python يقوم بتحويل رقم ثنائي إلى رقم عشري:

١. تعريف الدالة:`binary_to_decimal`

- تعريف دالة تأخذ رقمًا ثنائيًا كمدخل.
- يتم تهيئة المتغير `decimal` بقيمة صفر لتخزين الرقم العشري الناتج.

٢. حلقة:`for`

- يتم تنفيذ حلقة `for` لكل رقم ثنائي في الرقم الثنائي المدخل.
- يتم التحقق مما إذا كان الرقم غير '٠' أو '١'، في حالة وجود أي رقم غير صحيح يتم إرجاع.`None`

٣. تحويل الرقم الثنائي إلى عشري:

- يتم تحويل الرقم الثنائي إلى عشري عبر العمليات الرياضية المناسبة وتخزين النتيجة في المتغير.`decimal`

٤. إرجاع القيمة العشرية:

- يتم إرجاع القيمة العشرية المحسوبة.

٥. طلب إدخال رقم ثنائي من المستخدم:

- يُطلب من المستخدم إدخال رقم ثنائي.

٦. تحويل الرقم الثنائي إلى عشري والطباعة.

- يتم استدعاء الدالة `binary_to_decimal` مع الرقم الثنائي الذي أدخله المستخدم وتخزين القيمة المُحسوبة في `decimal_num`.

- يتحقق الشرط إذا كان `decimal_num` ليس`None` ، في حالة صحية يتم طباعة الرقم الثنائي والعشري المكافئ له، وإلا يتم طباعة "مدخل غير صالح."

باختصار، الكود يقوم بتحويل رقم ثنائي إلى رقم عشري ويتحقق من صحة الإدخال.

Question 3: Working with Files" Quiz Program"

Type python quiz program that takes a text or json or csv file as input for (20 (Questions, Answers)). It asks the questions and finally computes and prints user results and store user name and result in separate file csv or json file.

```python
import json
questions = { }
scores = 0
number=1
f = open("questions.txt",'r')
questions = json.load(f)
f.close()
print("python quiz programm")
print("Enter t for True or f for False")
name = input("Enter your full name: ")
for ques in questions.keys():
    print("Question",number,": ", ques)
    ans = input("The answer is ")
    if ans.upper() == questions[ques].upper():
        scores = scores + 1
        print("Correct ")
    else:
        print ("Wrong")
    number = number + 1
result={name:scores}
m = open("score.txt",'w')
result = json.dump(result,m)
m.close()
```

for ques in questions.keys()

questions - Notepad

File Edit Format View Help

```
{
"153.16.2.8 is a private ip address. ":"f",
"ARP refers to Address Resolution Protocol. ":"t",
"TCP is a network layer protocol. ":"f",
"ARP request message is a unicast message. ":"f",
"IPv4 is a 128-bit address. ":"f",
"IPv6 is a 128-bit address. ":"t",
"SDN refers to Software Defined Network. ":"t",
"10.0.0.5 is a private ip address. ":"t",
"UDP is a Transport Layer protocol. ":"t",
"224.0.0.9 is a multicast address. ":"t",
"Python is a machine language. ": "f",
"130.130.130.130 is a class C address. ":"f",
"MAC is address is 6 byte address. ":"t",
"IPv4 is a 32-bit address. ":"t",
"192.168.1.1 is a class A address. ":"f",
"IP is a network Layer protocol. ":"t",
"OSPF is a Routing Protocol. ":"t",
"ICMP refers to Internet Control Message Protocol. ":"t",
"hub is a layer 2 device . ":"f",
"bridge is a layer 3 device. ":"f"}
```

score - Notepad

File Edit Format View Help

```
{"Siba Mustafa": 9}
```

هذا الكود في Python هو برنامج اختبار بسيط يقوم بعرض سلسلة من الأسئلة من ملف نصي، يتيح للمستخدم إدخال إجاباته ويقوم بتقييم الإجابات الصحيحة. إليك شرح مفصل لكل جزء من الكود:

١ .يتم استيراد مكتبة JSON للتعامل مع تنسيقJSON .

٢ .تهيئة متغيرات:

:`questions` - قاموس يحتوي على الأسئلة والإجابات.

:`scores` - متغير لتتبع عدد الإجابات الصحيحة.

:`number` - متغير لتتبع رقم السؤال.

٣ .يتم فتح ملف "questions.txt" بوضع القراءة.

٤ .يتم تحميل الأسئلة من الملف "questions.txt" باستخدام `json.load(f)` وتخزينها في المتغير.`questions`

٥ .يتم إغلاق الملف بعد الانتهاء من استخدامه.

٦ .يتم طباعة رسالة ترحيبية للمستخدم.

٧ .يُطلب من المستخدم إدخال اسمه الكامل.

٨ .يتم عرض الأسئلة وطلب الإجابة من المستخدم بحلقة:`for`

- يتم عرض السؤال وطلب إدخال الإجابة.
- يتم التحقق من صحة الإجابة وزيادة `scores` إذا كانت صحيحة.
- يتم طباعة "Correct" إذا كانت الإجابة صحيحة، وإلا يتم طباعة."Wrong"
- يتم زيادة رقم السؤال.

٩ .يتم إنشاء قاموس يحتوي على اسم المستخدم وعدد الإجابات الصحيحة.

١٠ .يتم فتح ملف "score.txt" بوضع الكتابة.

١١ .يتم كتابة نتيجة الاختبار في الملف "score.txt" باستخدام`json.dump(result, m)` ، ثم يتم إغلاق الملف.

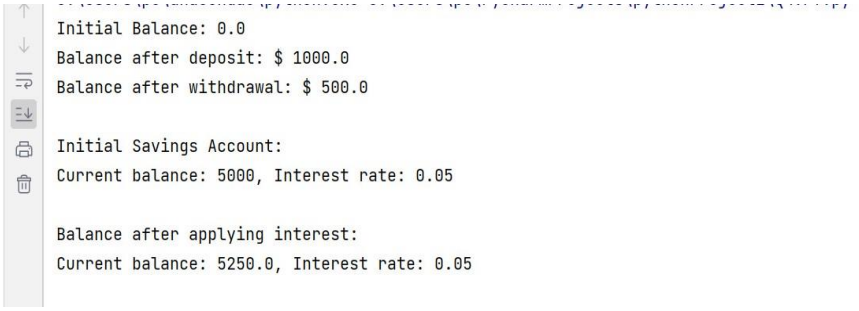
باختصار، البرنامج يقوم بإجراء اختبار للمستخدم باستخدام الأسئلة الموجودة في ملف نصي، ويقيم إجابات المستخدم.

Question 4 : Object-Oriented Programming - Bank Class

Define a class BankAccount with the following attributes and methods:
Attributes: account_number (string), account_holder (string), balance (float, initialized to 0.0)
Methods:deposit(amount), withdraw(amount) , get_balance()
- Create an instance of BankAccount, - Perform a deposit of $1000, - Perform a withdrawal of $500.
- Print the current balance after each operation.
- Define a subclass SavingsAccount that inherits from BankAccount and adds interest_rate Attribute and apply_interest() method that Applies interest to the balance based on the interest rate.
And Override print() method to print the current balance and rate.
- Create an instance of SavingsAccount , and call apply_interest() and print() functions.

```python
class BankAccount:
    def __init__(self, account_number, account_holder, balance=0.0):
        self.account_number = account_number
        self.account_holder = account_holder
        self.balance = balance

    def deposit(self, amount):
        self.balance += amount
        return self.balance

    def withdraw(self, amount):
        if amount <= self.balance:
            self.balance -= amount
            return self.balance
        else:
            return "Insufficient funds"

    def get_balance(self):
        return self.balance
```

```python

bank_account = BankAccount("1357", "AHMAD")
print("Initial Balance:", bank_account.get_balance())
bank_account.deposit(1000)
print("Balance after deposit: $", bank_account.get_balance())
bank_account.withdraw(500)
print("Balance after withdrawal: $", bank_account.get_balance())

savings_account= SavingsAccount("2468", "OSAMA", 5000, 0.05)
print("\nInitial Savings Account:")
savings_account.print_details()

savings_account.apply_interest()
print("\nBalance after applying interest:")
savings_account.print_details()
```



```
Initial Balance: 0.0
Balance after deposit: $ 1000.0
Balance after withdrawal: $ 500.0

Initial Savings Account:
Current balance: 5000, Interest rate: 0.05

Balance after applying interest:
Current balance: 5250.0, Interest rate: 0.05
```

الكود يعرض استخدام للوراثة في Python من خلال تعريف فئتين `BankAccount` و`SavingsAccount`، حيث ترث `SavingsAccount`من:`BankAccount`

١. فئة:`BankAccount`
- تحتوي على الخصائص التالية:
`account_number`: - رقم الحساب.
`account_holder`: - اسم صاحب الحساب.
`balance`: - الرصيد الحالي للحساب.
- تحتوي على الدوال التالية:
`__init__`: - دالة البناء تهيئ الحساب برقم الحساب، اسم صاحب الحساب، والرصيد الابتدائي.
`deposit`: - دالة لإيداع الأموال إلى الحساب.
`withdraw`: - دالة لسحب الأموال من الحساب مع التحقق من توفر الأموال.
`get_balance`: - دالة للحصول على الرصيد الحالي للحساب.

٢. فئة) `SavingsAccount` ترث من:(`BankAccount`
- تحتوي على الخصائص التالية:
`interest_rate`: - معدل الفائدة لحساب التوفير.
- تحتوي على الدوال التالية:
`__init__`: - دالة البناء تهيئ حساب التوفير بالمتغيرات الأساسية ومعدل الفائدة.
`apply_interest`: - دالة لحساب الفائدة وإضافتها إلى الرصيد.
`print_details`: - دالة لطباعة تفاصيل حساب التوفير.

٣. في الجزء الأخير من الكود:
- يتم إنشاء حساب بنكي `bank_account` وعرض التعامل مع الإيداع والسحب.
- يتم إنشاء حساب توفير `savings_account` وعرض طباعة التفاصيل الأولية وتطبيق الفائدة.

باختصار، الكود يوضح كيفية استخدام الوراثة في Python لإنشاء هيكلية تسمح بإعادة استخدام السمات والسلوكيات بين الفئات المرتبطة.