



Name: حيدرة علي حامد, Number: 2246, Submitted To GitHub: <https://github.com/HaedaraHamed/First-Network-Programming-Homework>

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}`

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

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.

Tips: using loop, `'len()'`, `startswith()` methods.

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}`

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. **Tips:** solve input errors.

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.

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.

Notes "important"

- Homework is accepted as **well explained Pdf & "Nicely Formatted Code"** "You can do all job in one notebook then print as pdf or "copy and paste" on word document "use" then convert into pdf with extra info " - You have to show:

Question number >> Question itself >> your answer code with explanations > your Result "you can use this doc as template"

- You Have to Show code execution as Screenshots from your laptop or phone".

- Apply your full name and number, Homework number to pdf.

- **Similar Solutions** will **rejected** and not accepted.

- The Homework is accepted until the date of **"27/5/2024"**, if after >> **mark=mark- (current_date - 27/5/2024)*0.3** - upload your code to your **GitHub Account**, **"PDF + Code"**

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}
```

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

```
x = int(input("Enter your number: "))
y = 1

if x < 0:
    print("Sorry!!")
elif x == 0:
    print("1")
else:
    for i in range(1, x + 1):
        y *= i

    print(f" {x}! = {y}")
```

```
Enter your number: 3
3! = 6
```

توضيح:

الكود السابق يقوم بحساب عامل الضرب للعدد المدخل x ويعرض النتيجة:

1. يتم قراءة العدد من المستخدم باستخدام `input()`.
 2. يتم تعيين قيمة ابتدائية للمتغير `y` بقيمة 1.
 3. يتم التحقق من قيمة `x`:
 - إذا كانت أقل من صفر، يتم عرض "Sorry!!".
 - إذا كانت تساوي صفر، يتم عرض "1".
 - في حالة أخرى، يتم استخدام حلقة `for` لحساب عامل الضرب للعدد `x`.
 4. يتم عرض النتيجة باستخدام `print()`.
- بالنسبة لحساب عامل الضرب، يتم ضرب الأعداد من 1 إلى `x` معًا. مثلاً، إذا كان `x = 5`، فإن $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

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

```
L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']
L1 = []
for i in L:
    if i.startswith('B'):
        L1.append(i)
print("Items starting with B:", L1)
```

➡ Items starting with B: ['Bio']

توضيح:

- يتم إنشاء قائمة `L` تحتوي على خمس سلاسل نصية 'Network'، 'Bio'، 'Programming'، 'Physics' و 'Music'.
- يتم إنشاء قائمة فارغة `L1` لتخزين العناصر المصفاة.
- يتم استخدام حلقة `for` للتكرار عبر كل عنصر في القائمة `L`.
- داخل الحلقة، يتم استخدام جملة `if` للتحقق مما إذا كان العنصر الحالي يبدأ بالحرف 'B' باستخدام طريقة `startswith()`.
- إذا تحققت الشرط، يتم إضافة العنصر إلى القائمة `L1` باستخدام طريقة `append()`.
- بعد معالجة جميع العناصر، يتم طباعة "العناصر التي تبدأ بـ 'B': تليها العناصر المصفاة في القائمة `L1`. في هذا الحالة، يتم طباعة 'Bio' فقط

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}`

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

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

توضيح:

يتم استخدام الكود السابق لإنشاء قاموس يحتوي على مفاتيح من 0 إلى 10، وقيمها تكون مساوية للمفتاح + 1.

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. **Tips:** solve input errors.

```
[ ] x = input("Enter your binary number: ")  
    try:  
        y = int(x, 2)  
        print(f"{x} = {y}")  
    except ValueError:  
        print("Please enter a valid binary number.")
```

```
⇒ Enter your binary number: 01100  
   01100 = 12
```

توضيح:

- يبدأ البرنامج بطلب إدخال رقم ثنائي من المستخدم باستخدام `input()`.
- يتم تخزين القيمة المدخلة في المتغير `x`.
- يتم استخدام كتلة `try` و `except` للتعامل مع الأخطاء المحتملة.
- داخل الكتلة، يتم استخدام `int(x, 2)` لتحويل القيمة المدخلة (الرقم الثنائي) إلى صيغة عشرية.
- إذا نجح التحويل، يتم طباعة القيمة الثنائية والعشرية معًا.
- إذا كانت القيمة المدخلة غير صحيحة (مثل عدم كونها رقمًا ثنائيًا صحيحًا)، يتم طباعة رسالة خطأ.

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.

في البداية سنقوم بإنشاء ملف csv يحتوي (20 سؤال و جواب)) وحفظه باسم quiz.csv من خلال الكود التالي:

```
import csv

questions_and_answers = [
    ("دمشق", "ما هي عاصمة سورية"),
    ("بغداد", "ما هي عاصمة العراق"),
    ("بيروت", "ما هي عاصمة لبنان"),
    ("القدس", "ما هي عاصمة فلسطين"),
    ("الجزائر", "ما هي عاصمة الجزائر"),
    ("القاهرة", "ما هي عاصمة مصر"),
    (" صنعاء", "ما هي عاصمة اليمن"),
    ("المنامة", "ما هي عاصمة البحرين"),
    ("الكويت", "ما هي عاصمة الكويت"),
    ("طرابلس الغربية", "ما هي عاصمة ليبيا"),
    ("الرياض", "ما هي عاصمة السعودية"),
    ("تونس", "ما هي عاصمة تونس"),
    (14, "كم عدد المحافظات السورية"),
    (6, "كم عدد الجامعات الحكومية في سورية"),
    ("روما", "ما هي عاصمة إيطاليا"),
    ("برلين", "ما هي عاصمة ألمانيا"),
    ("باريس", "ما هي عاصمة فرنسا"),
    ("بيكين", "ما هي عاصمة الصين"),
    ("موسكو", "ما هي عاصمة روسيا"),
    (22, "كم عدد الدول العربية"),
]

csv_file_path = "quiz.csv"
with open(csv_file_path, mode="w", newline="") as file:
    writer = csv.writer(file)
    writer.writerows(questions_and_answers)

print(f"CSV file '{csv_file_path}' created successfully.")
```

→ csv file 'quiz.csv' created successfully.

فيكون ملف csv الذي قمنا بإنشائه بهذا الشكل:

quiz.csv	haedara	...
1 to 19 of 19 entries Filter 		
ماهي عاصمة سورية	دمشق	
ماهي عاصمة العراق	بغداد	
ما هي عاصمة لبنان	بيروت	
ما هي عاصمة فلسطين	القدس	
ما هي عاصمة الجزائر	الجزائر	
ما هي عاصمة مصر	القاهرة	
ما هي عاصمة اليمن	صنعاء	
ما هي عاصمة البحرين	المنامة	
ما هي عاصمة الكويت	الكويت	
ما هي عاصمة ليبيا	طرابلس الغربية	
ما هي عاصمة السعودية	الرياض	
ما هي عاصمة تونس	تونس	
كم عدد المحافظات السورية	14	
كم عدد الجامعات الحكومية في سورية	6	
ماهي عاصمة ايطاليا	روما	
ما هي عاصمة المانيا	برلين	
ما هي عاصمة فرنسا	باريس	
ما هي عاصمة الصين	بكين	
ما هي عاصمة روسيا	موسكو	
كم عدد الدول العربية	22	
Show 25 per page		

والان سوف نقوم بكتابة الكود الذي يطرح الأسئلة وأخيراً يحسب ويطبع نتائج المستخدم ويخزن اسم المستخدم وينتج عنه ملف منفصل بتنسيق CSV



```
import csv
import random

def load_quiz_data(quiz):

    with open(quiz, "r") as quiz_file:
        reader = csv.reader(quiz_file)
        quiz_data = list(reader)
    return quiz_data

def ask_questions(quiz_data):

    score = 0
    questions_right = 0
    question_no = 1

    random.shuffle(quiz_data)
    for data in quiz_data:
        question, correct_answer = data
        print(f"Question #{question_no}: {question}")
        user_answer = input("What is your answer? ")

        if user_answer.lower() == correct_answer.lower():
            print("Correct.")
            score += 1
            questions_right += 1
        else:
            print("Incorrect.")

        question_no += 1

    total_score = (score / len(quiz_data)) * 100
    return questions_right, total_score

def main():
    file_name = input(" enter the name of the quiz file (CSV format): ")
    quiz_data = load_quiz_data(file_name)
    user_name = input(" enter your name: ")
    questions_right, total_score = ask_questions(quiz_data)
    print(f"You got {questions_right} questions right= {total_score:.2f}%.")

    with open(user_name, "w", newline="") as result_file:
        writer = csv.writer(result_file)
        writer.writerow(["Name", "Questions Right", "Score"])
        writer.writerow([user_name, questions_right, total_score])

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

توضيح في الكود السابق:

1. `load_quiz_data(quiz)` :

- هذه الدالة تقوم بتحميل بيانات الاختبار من ملف CSV.
- تفترض أن تنسيق الملف هو "question,answer".
- تقوم بقراءة البيانات من الملف وتخزينها في قائمة.
- تعيد قائمة الأسئلة والإجابات.

2. `ask_questions(quiz_data)` :

- هذه الدالة تطرح الأسئلة على المستخدم وتحسب النتائج.
- تقوم بتبديل ترتيب الأسئلة بشكل عشوائي.
- تعرض كل سؤال وتساءل المستخدم عن إجابته.

○ تحسب النتيجة الإجمالية وتعيدها.

3. Main() :

- تطلب من المستخدم اسم ملف الاختبار مثل ("quiz.csv")
- تحمل بيانات الاختبار باستخدام load_quiz_data
- تطرح الأسئلة باستخدام ask_questions
- تعرض النتيجة الإجمالية.
- تخزن اسم المستخدم والنتيجة في ملف منفصل

خرج تنفيذ الكود السابق:



enter the name of the quiz file (CSV format): quiz.csv

enter your name: haedara

Question #1: ما هي عاصمة الصين

What is your answer? بكين

Correct.

Question #2: ما هي عاصمة ليبيا

What is your answer? ليبيا

Incorrect.

Question #3: ما هي عاصمة فلسطين

What is your answer? القدس

Correct.

Question #4: ما هي عاصمة الكويت

What is your answer? الكويت

Correct.

Question #5: كم عدد الجامعات الحكومية في سورية

What is your answer? 6

Correct.

Question #6: ما هي عاصمة تونس

What is your answer? تونس

Correct.

Question #13: ما هي عاصمة لبنان

What is your answer? بيروت

Correct.

Question #14: كم عدد المحافظات السورية

What is your answer? 14

Correct.

Question #15: ما هي عاصمة إيطاليا

What is your answer? روما

Correct.

Question #16: ما هي عاصمة العراق

What is your answer? بغداد

Correct.

Question #17: كم عدد الدول العربية

What is your answer? 22

Correct.

Question #18: ما هي عاصمة روسيا

What is your answer? موسكو

Correct.

Question #19: ما هي عاصمة فرنسا

What is your answer? باريس

Correct.

Question #20: ما هي عاصمة اليمن

What is your answer? صنعاء

Correct.

You got 19 questions right= 95.00%.


```
quiz.csv      haedara X  ...
1 Name,Questions Right,Score
2 haedara,19,95.0
3
```

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.



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

    def deposit(self, amount):
        self.balance += amount
        print(f"Amount Deposited: ${amount:.2f}")

    def withdraw(self, amount):
        if self.balance >= amount:
            self.balance -= amount
            print(f"You Withdrew: ${amount:.2f}")
        else:
            print("Insufficient balance")

    def get_balance(self):
        return self.balance

class SavingsAccount(BankAccount):
    def __init__(self, account_number, account_holder, interest_rate):
        super().__init__(account_number, account_holder)
        self.interest_rate = interest_rate

    def apply_interest(self):
        interest_amount = self.balance * (self.interest_rate / 100)
        self.balance += interest_amount

    def print(self):
        print(f"Current Balance: ${self.balance:.2f}")
        print(f"Interest Rate: {self.interest_rate}%")

bank_account = BankAccount(account_number="123456", account_holder="haedara")
bank_account.deposit(1000)
bank_account.withdraw(500)
print(f"Current Balance: ${bank_account.get_balance():.2f}")

savings_account = SavingsAccount(account_number="789012", account_holder="noor", interest_rate=2.5)
savings_account.deposit(2000)
savings_account.apply_interest()
savings_account.print()
```

نلاحظ خرج تنفيذ الكود السابق:



```
Amount Deposited: $1000.00
You Withdrew: $500.00
Current Balance: $500.00
Amount Deposited: $2000.00
Current Balance: $2050.00
Interest Rate: 2.5%
```

توضيح للكود السابق:

1. BankAccount :

- هذا الكلاس يمثل حسابًا مصرفيًا عامًا.
- يحتوي على الخصائص التالية:
 - `account_number` :رقم الحساب.
 - `account_holder` : اسم صاحب الحساب.
 - `Balance` :الرصيد الحالي للحساب (الذي يبدأ بالقيمة 0.0).
- يحتوي على ثلاثة أساليب:
 - `deposit (amount)` : يقوم بإيداع المبلغ المحدد في الحساب.
 - `withdraw (amount)` : يقوم بسحب المبلغ المحدد من الحساب إذا كان الرصيد كافيًا.
 - `get_balance ()` : يعيد الرصيد الحالي للحساب.

2. SavingsAccount :

- هذا الكلاس يمثل حسابًا مصرفيًا للتوفير.
- يرث من `BankAccount` ويضيف خاصية إضافية:
 - `interest_rate` :معدل الفائدة للحساب.
- يحتوي على أسلوب واحد:
 - `apply_interest ()` : يطبق الفائدة على الرصيد استنادًا إلى معدل الفائدة.
- `print ()` : يعرض الرصيد الحالي ومعدل الفائدة.

1. يتم إنشاء حساب مصرفي باسم "haedara" ورقم حساب "123456".
2. يتم إيداع 1000 دولار في الحساب.
3. يتم سحب 500 دولار من الحساب (إذا كان الرصيد كافيًا).
4. يتم عرض الرصيد الحالي للحساب.
5. يتم إنشاء حساب توفير باسم "noor" ورقم حساب "789012" ومعدل فائدة 2.5%.
6. يتم إيداع 2000 دولار في الحساب التوفير.
7. يتم تطبيق الفائدة على الرصيد.
8. يتم عرض الرصيد الحالي ومعدل الفائدة للحساب التوفير.