



الجمهورية العربية السورية

جامعة تشرين

قسم هندسة الاتصالات والالكترونيات

السنة الخامسة

مادة برمجة الشبكات

اعداد الطالب:

لجين أحمد مهنا 2894

بأشراف الدكتور:

مهند عيسى

Question 1:

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 }

الكود:

[7]:

```
new*  
1 L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']  
2 L2 = [80, 443, 21, 53]  
3  
4 d = {}  
5 for i in range(len(L1)):  
6     d[L1[i]] = L2[i]  
7  
8 print(d)  
9
```

التفسير:

الشرح:

1. تعريف قائمتين L1 و L2 كما في السؤال .

2. تعريف متغير فارغ d كقاموس.

3. استخدام حلقة for للمرور على عناصر القائمة L1. يتم استخدام دالة len() من خلالها نحصل على عدد العناصر في القائمة .

٤. داخل حلقة for، يتم إضافة عنصر من قائمة L1 كمفتاح في القاموس d ويتم ربطه بعنصر مقابله من قائمة L2 باستخدام الفهرس i .

٥. أخيرا يتم طباعة القاموس d **النتيجة:**

```
← TAB _ ⋮
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
[Program finished]
```

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

الكود:

```
≡ new* Ⓛ ⚡ ⋮
1 def fact (n):
2     if n < 0:
3         return "Factorial does not exist "
4     elif n == 0:
5         return 1
6     else:
7         return n * fact (n - 1)
8
9 num = int(input("Enter a number: "))
10 result = fact (num)
11 print(result)
```

التفسير:

1. قمت بتعريف دالة اسمها fact تستقبل عدد صحيح n كمعامل وتبدأ بفحصه

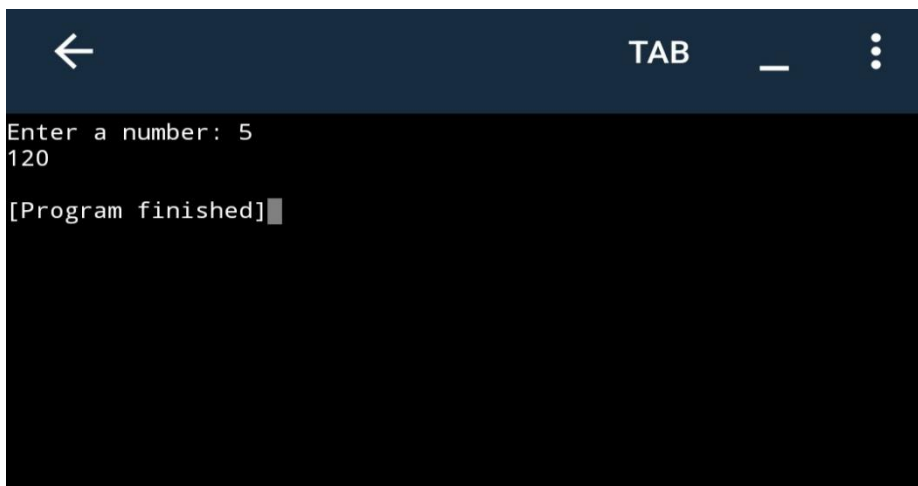
_ إذا كان أصغر من الصفر يعيد جملة factorial does not exist

_ إذا كان يساوي الصفر تعيد الدالة fact القيمة 1 أو تعيد

$n * \text{fact}(n-1)$

2. قمت باستدعاء الدالة يطلب من المستخدم إدخال عدد صحيح يتم استخدام هذا العدد كمعامل لدالة fact لنحسب عامليته ثم يتم طباعة النتيجة .

النتيجة:



```
Enter a number: 5
120
[Program finished]
```

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 prin .

الكود:

```
new*  
1 L = ['Network', 'Bio', 'Programming', 'Physics',  
2 'Music']  
3 for i in range(len(L)):  
4     if L[i].startswith('B'):  
        print(L[i])
```

التفسير:

1. عرفت قائمة L الموجودة في السؤال
2. استخدمت الحلقة for للتكرار على عناصر الموجودة ضمن القائمة L.
3. تحققت باستخدام التابع if إذا كانت الكلمة تبدأ بحرف B .
4. طبعت النتيجة والتي هي Bio .

النتيجة:

```
← TAB _ ⋮  
Bio  
[Program finished]
```

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}`

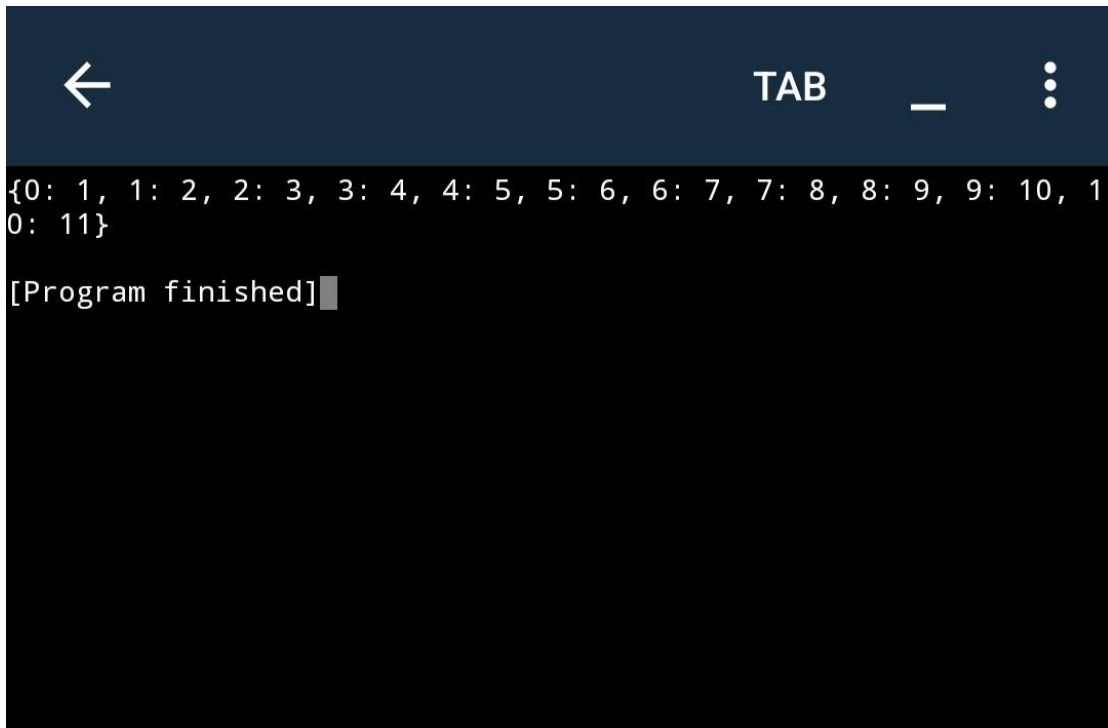
الكود:

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

التفسير:

1. سينشئ هذا الكود قاموساً `d` حيث يكون المفتاح هو العدد من 0 إلى 10 والقيمة هي العدد التالي بعد المفتاح .

النتيجة:



```
← TAB _ ⋮
{0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}
[Program finished]
```

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.

الكود:

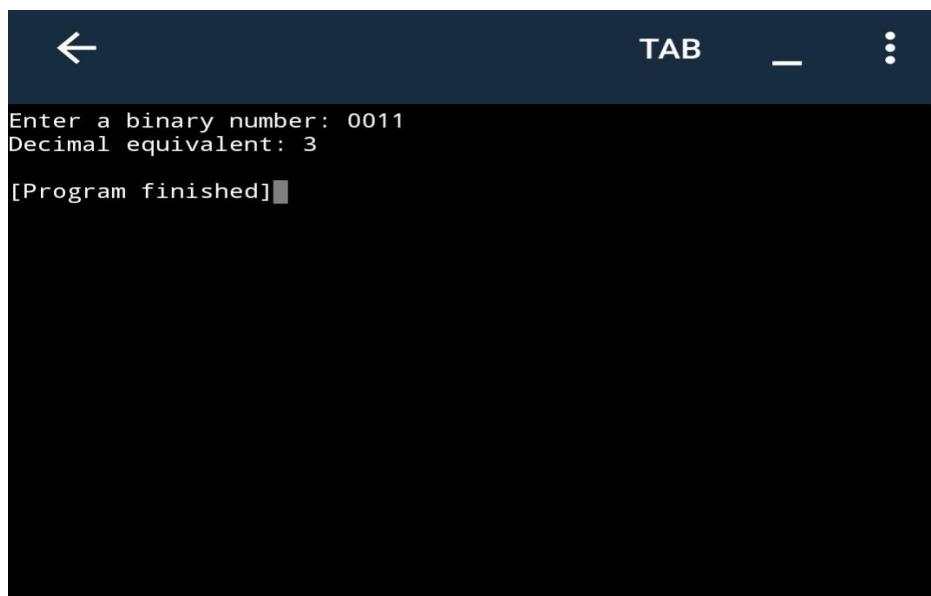
```
new*  
1 binary = input("Enter a binary number: ")  
2 if not binary.isdigit():  
3     print("Invalid input. Please enter a binary  
   number.")  
4 else:  
5     decimal = int(binary, 2)  
6     print("Decimal equivalent:", decimal)
```

التفسير:

1. نستخدم دالة input() لقراءة القيمة المدخلة من المستخدم وتخزينها في المتغير
2. binary. نستخدم الشرط if للتحقق من صحة القيمة المدخلة باستخدام دالة .isdigit()

_ إذا كانت القيمة المدخلة غير صحيحة، سيتم طباعة رسالة خطأ. إلا إذا كانت القيمة المدخلة صحيحة، سنستخدم دالة int() لتحويلها إلى عدد صحيح ونستخدم البادئة 2 للإشارة إلى أن العدد هو بنية ثنائية. ثم نطبع القيمة العشرية المكافئة على الشاشة. تم تجربة الرقم ٠٠١١ وأعيدت القيمة ٣ .

النتيجة:



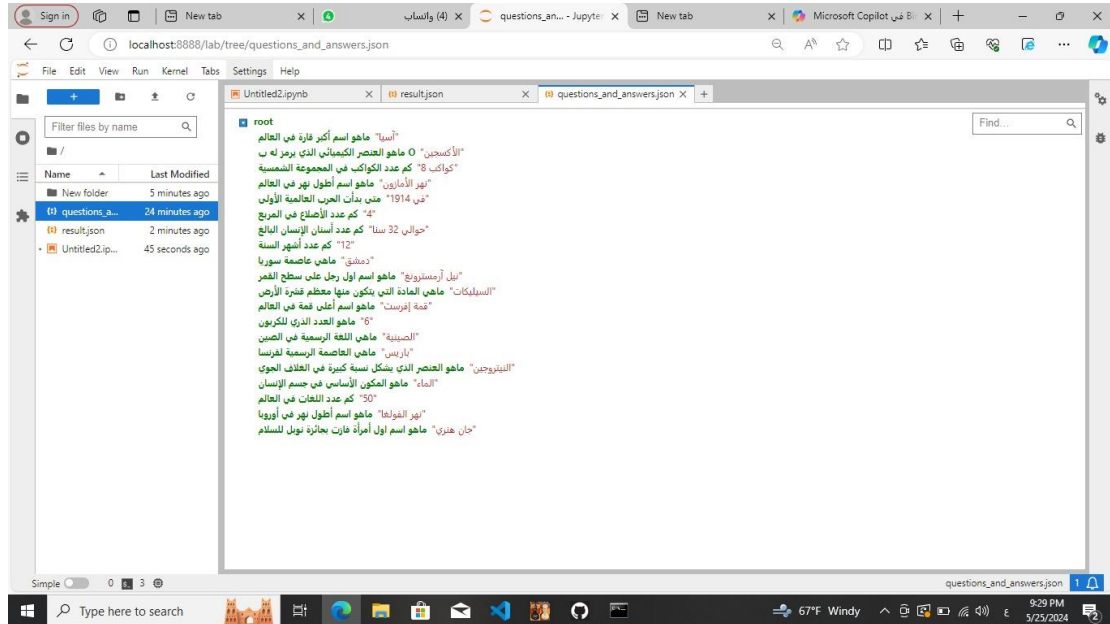
```
Enter a binary number: 0011
Decimal equivalent: 3
[Program finished]
```

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

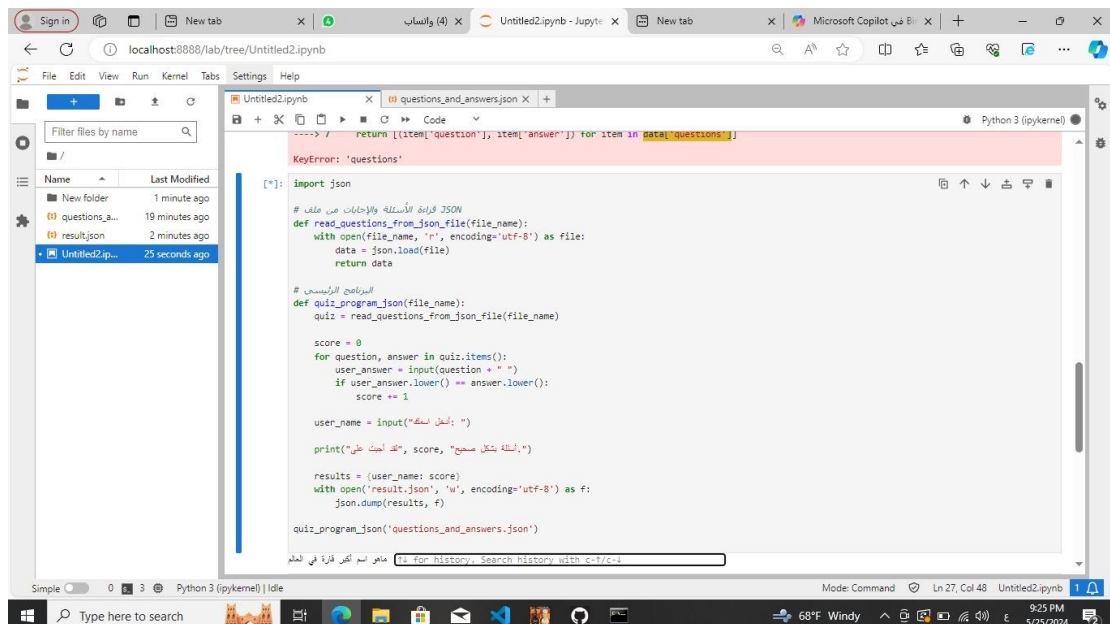
الكود:

1. تم كتابة 20 سؤالاً مع جواباً في ملف وحفظ بامتداد json حيث البيانات على هيئة قاموس فكان السؤال هو مفتاح القاموس والجواب هو قيمة المفتاح:



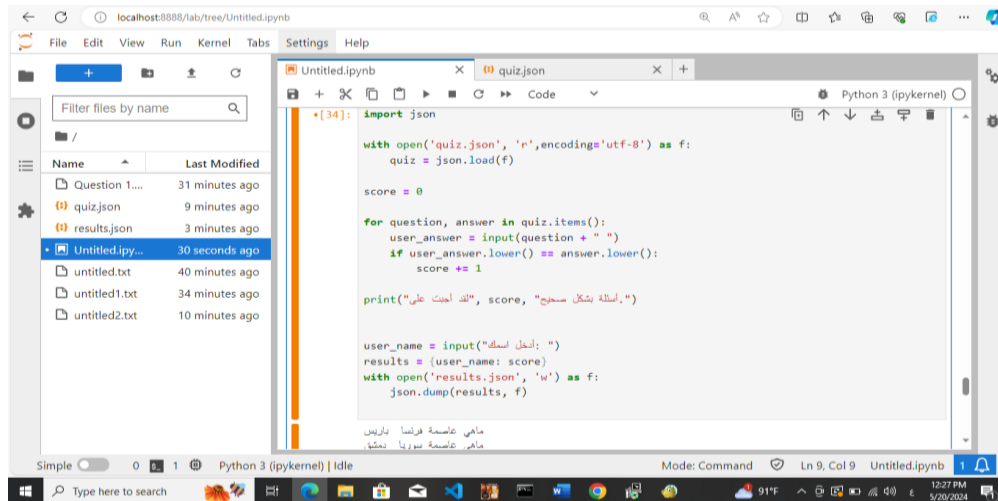
2.

- استيراد المكتبة json التي تتعامل مع ملفات بامتداد .json.



- تم تشغيل الملف quiz.json بوضع القراءة 'r' ونمط الترميز 'utf-8'.
- يتم قراءة الملف وتخزين البيانات كقاموس في المتغير quiz.
- يتم انشاء المتغير score لتتبع عدد الأسئلة التي أجاب عليها المستخدم بشكل صحيح.

- يتم تنفيذ الحلقة for تتكرر على كل سؤال وجواب في القاموس quiz.
- في كل تكرار يتم طرح السؤال على المستخدم وينتظر البرنامج حتى الإجابة.
- بالنهاية يطلب من المستخدم ادخال اسمه ويتم انشاء قاموسا جديدا يحوي على اسم المستخدم وعدد النقاط التي حصل عليها.



```

[34]: import json

with open('quiz.json', 'r', encoding='utf-8') as f:
    quiz = json.load(f)

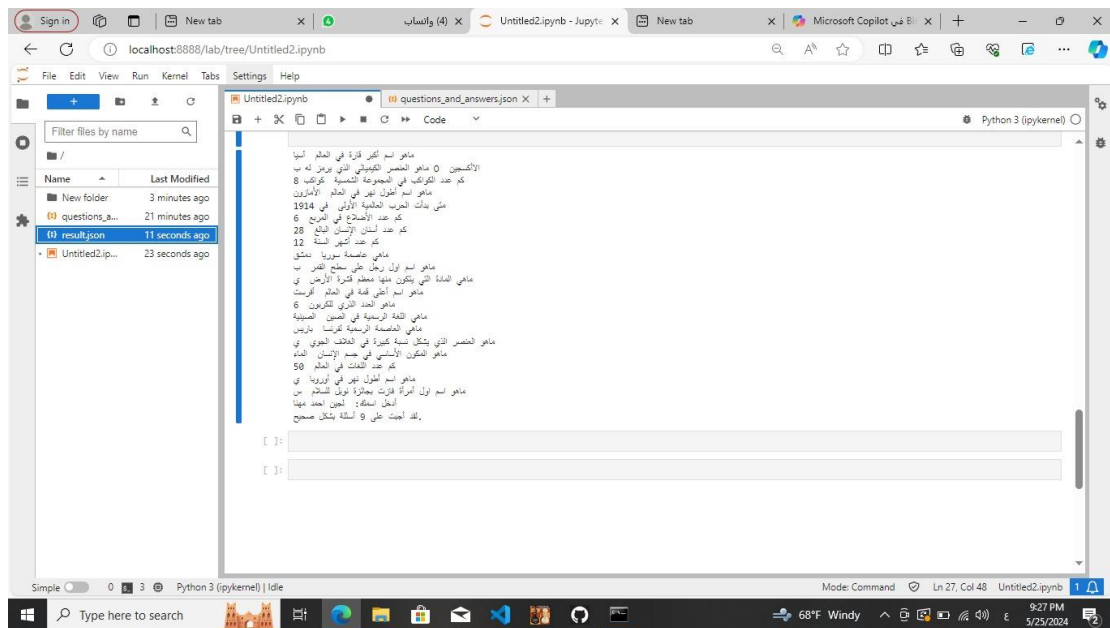
score = 0

for question, answer in quiz.items():
    user_answer = input(question + " ")
    if user_answer.lower() == answer.lower():
        score += 1

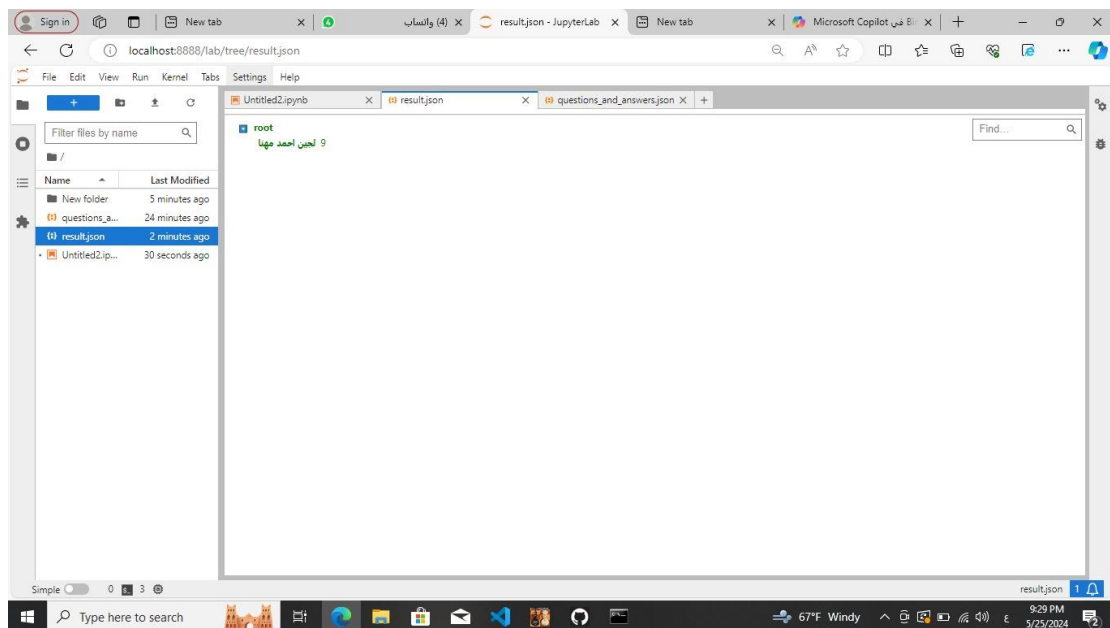
print("اسئلة بشكل صحيح", score, "لقد أجبت على")

user_name = input("ادخل اسمك: ")
results = {user_name: score}
with open('results.json', 'w') as f:
    json.dump(results, f)
  
```

3. النتيجة بعد الإجابة على الأسئلة كاملة.



ملف ال results الناتج عن تنفيذ الكود السابق حيث هذا الملف أيضا بامتداد json ويحوي بداخله على اسم المستخدم ونتيجته على الإجابة عن الأسئلة.



Question 4:

1. 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.

الكود:



new*



```
1 class BankAccount:
2     def __init__(self, account_number,
3         account_holder):
4         self.account_number = account_number
5         self.account_holder = account_holder
6         self.balance = 0.0
7     def deposit(self, amount):
8         self.balance += amount
9     def withdraw(self, amount):
10        if self.balance >= amount:
11            self.balance -= amount
12        else:
13            print("Insufficient funds")
14    def get_balance(self):
15        return self.balance
16 class SavingsAccount(BankAccount):
17     def __init__(self, account_number,
18         account_holder, interest_rate):
19         super().__init__(account_number,
20             account_holder)
21         self.interest_rate = interest_rate
22     def apply_interest(self):
23         interest_amount = self.balance * (self.
24             interest_rate / 100)
25         self.deposit(interest_amount)
26     def __str__(self):
27         return f"Balance: {self.balance}, Interest
28             Rate: {self.interest_rate}%"
29 # Create an instance of BankAccount
30 bank_acc = BankAccount("123456", "Iujain ")
31 bank_acc.deposit(1000)
32 print(bank_acc.get_balance())
33 bank_acc.withdraw(500)
34 print(bank_acc.get_balance())
35 # Create an instance of SavingsAccount
36 savings_acc = SavingsAccount("789", "Iujain ", 5)
```



Tab

:

;

'

#

(



new*



```
14         return self.balance
15     class SavingsAccount(BankAccount):
16         def __init__(self, account_number,
17             account_holder, interest_rate):
18             super().__init__(account_number,
19                 account_holder)
20             self.interest_rate = interest_rate
21
22         def apply_interest(self):
23             interest_amount = self.balance * (self.
24                 interest_rate / 100)
25             self.deposit(interest_amount)
26         def __str__(self):
27             return f"Balance: {self.balance}, Interest
28                 Rate: {self.interest_rate}%"
29
30     # Create an instance of BankAccount
31     bank_acc = BankAccount("123456", "Iujain ")
32     bank_acc.deposit(1000)
33     print(bank_acc.get_balance())
34     bank_acc.withdraw(500)
35     print(bank_acc.get_balance())
36
37     # Create an instance of SavingsAccount
38     savings_acc = SavingsAccount("789012",
39         "ahmad", 5)
40     savings_acc.deposit(2000)
41     savings_acc.apply_interest()
42     print(savings_acc)
```

التفسير:

قمت بتعريف فئتين حسب نص السؤال BankAccount و SavingsAccount.
أولا

1. BankAccount:

حساب بنكي بسيط.

- يحتوي على متغيرات account_number لتخزين رقم الحساب
account_holder لتخزين صاحب الحساب.

- يحتوي على balance_ لتخزين رصيد الحساب.

1. (self).balance def

لتحويل الدالة balance إلى خاصية قابلة للقراءة فقط.

- عند استدعاء obj.balance سيتم تنفيذ الدالة balance وإرجاع قيمة
الرصيد _balance.

2. `def deposit(self, amount):`

- هذه الدالة تقوم بإضافة مبلغ معين إلى الرصيد الحالي في الحساب.
- تقوم بزيادة قيمة المتغير `balance` بالمبلغ المحدد.

3. `def withdraw(self, amount):`

- هذه الدالة تقوم بسحب مبلغ معين من الرصيد الحالي في الحساب.
- تقوم بالتحقق مما إذا كان الرصيد كافيًا لسحب المبلغ المطلوب، وإذا كان كافيًا، يتم خصم المبلغ من الرصيد.
- إذا كان الرصيد غير كافٍ، يتم طباعة رسالة "Insufficient funds" لتنبيه صاحب الحساب بأنه لا يوجد رصيد كافٍ للسحب.

قمنا بتعريف الدالة `__init__` التي تهدف إلى تهيئة الكائن عند إنشائه وتعيين قيم ابتدائية للمتغيرات.

- يحتوي على الدوال `balance` لاسترجاع قيمة الرصيد، و `deposit` و `withdraw` لإضافة وسحب الأموال من الحساب.

2. `SavingsAccount:`

- تمثل حساب توفير يرث كل خصائص `BankAccount`.
- يحتوي على متغير إضافي `interest_rate` لتخزين معدل الفائدة على الحساب.

- يحتوي على دالة `apply_interest` التي تحسب المبلغ المستحق كفاية على الرصيد وتضيفها إلى الرصيد باستخدام دالة `deposit`.

الملخص : ترث فئة SavingsAccount من فئة BankAccount، مما يعني أنها تأخذ جميع الخصائص والسلوكيات من الفئة الأساسية وتضيف إليها خصائص إضافية.

النتيجة:

```
← TAB _ ⋮
1000.0
500.0
Balance: 2100.0, Interest Rate: 5%
[Program finished]
```