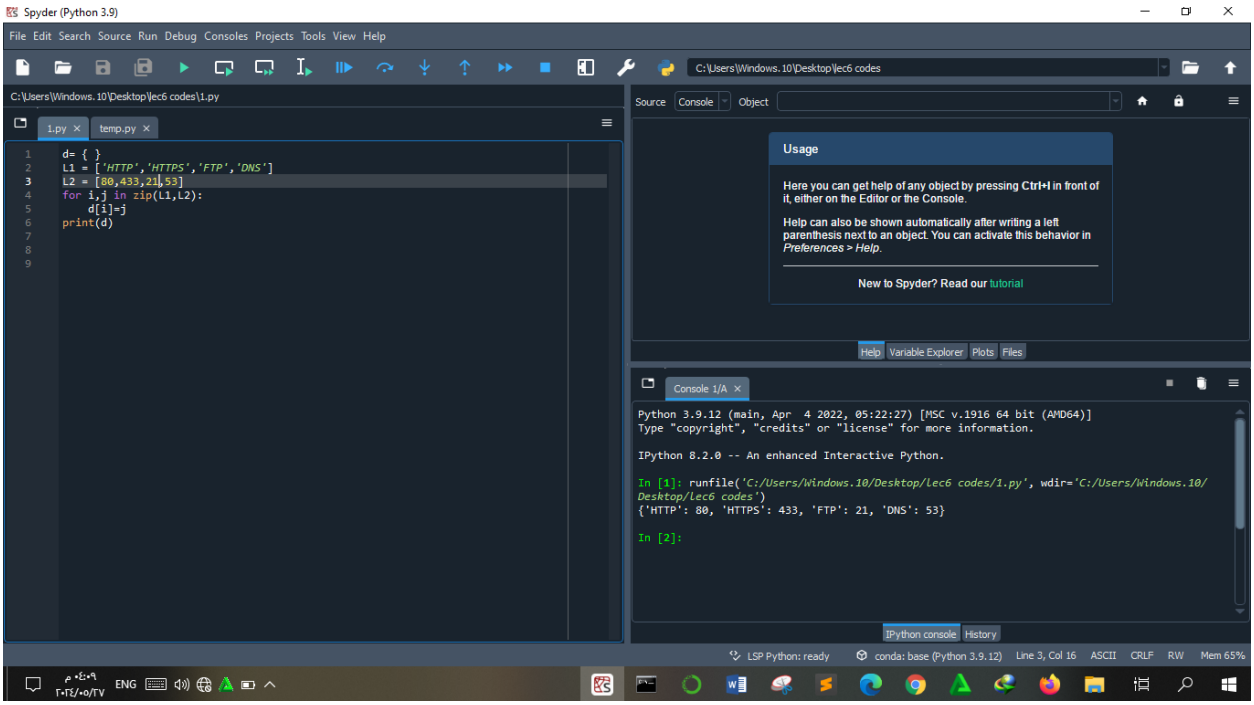
Name : GHAZAL HABIB

Number : 2424

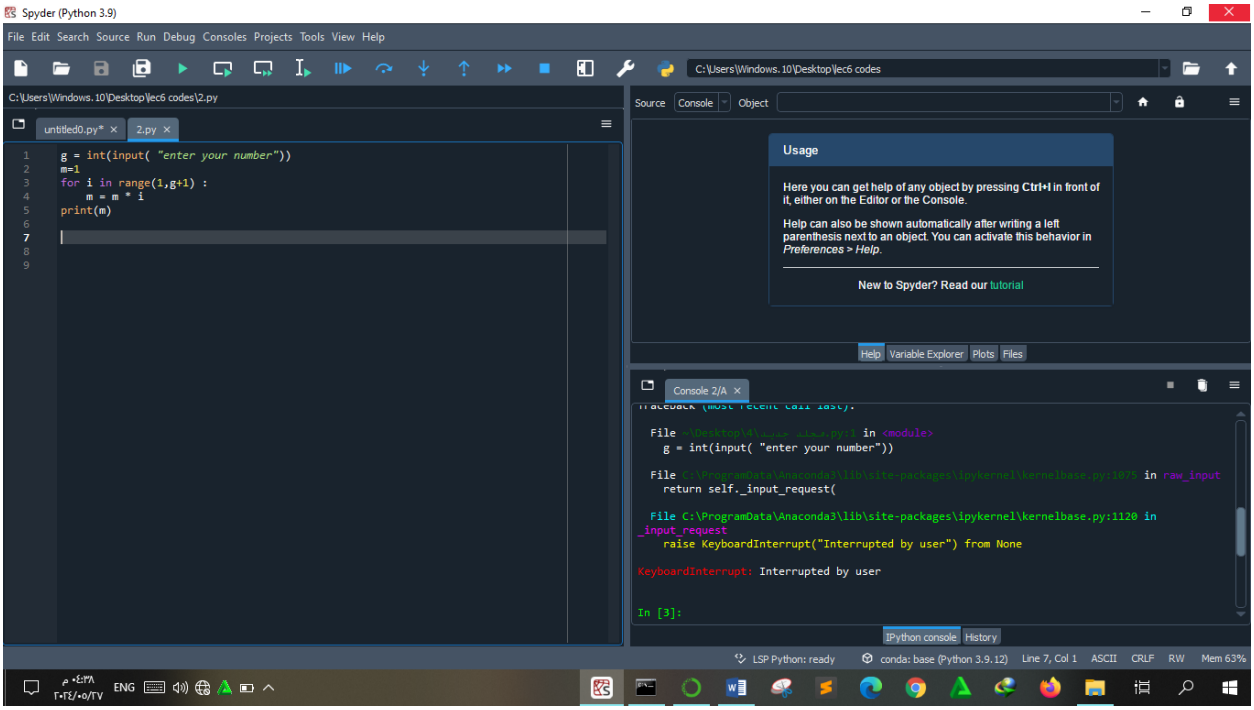
Submitted to GitHub :

Question 1:

A-If you have two lists, L1=[‘HTTP’,’HTTPS’,’FTP’,’DNS’] L2=[80,443,20,53], convert it to generate this dictionary d={‘HTTP’:80,’HTTPS’:443,’FTP’:21,’DNS’:53 }



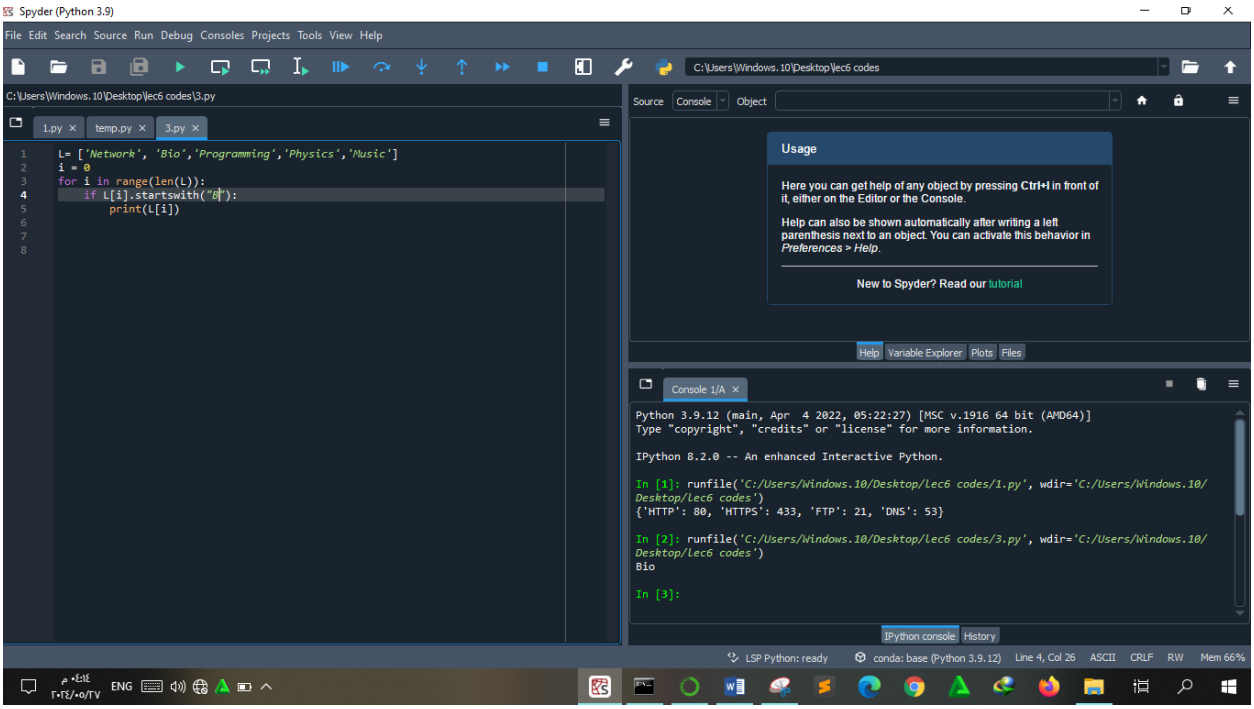
B- Write a Python program that calculates the factorial of a given number entered by user.



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

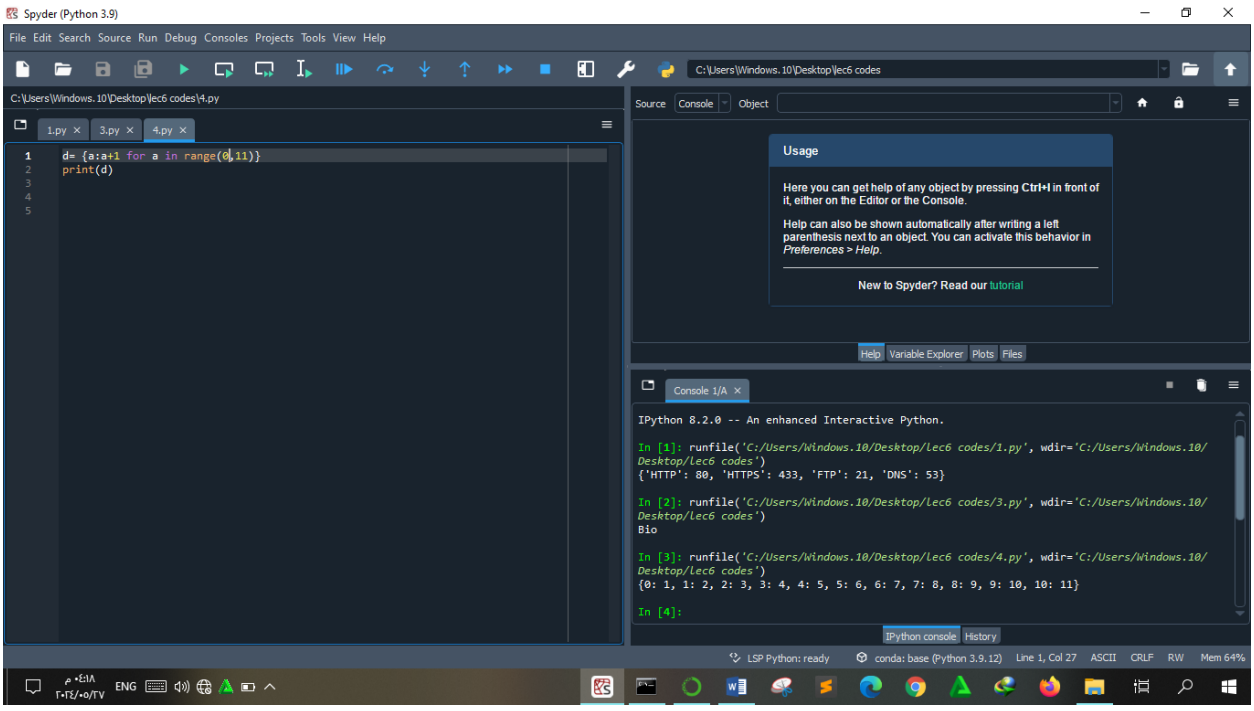
In this exercise, you will implement a Python program that reads the items of the previous list and identifies the items that starts with 'B' letter, then print it on screen.

Tips: using loop, 'Len ()' , starts with() method



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}


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Windows.10\Desktop\lec6 codes
C:\Users\Windows.10\Desktop\lec6 codes\4.py
1.py
3.py
4.py
d= {a:a+1 for a in range(0,11)}
print(d)
Source
Console
Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Help
Variable Explorer
Plots
Files
Console 1/A
IPython 8.2.0 -- An enhanced Interactive Python.
In [1]: runfile('C:/Users/Windows.10/Desktop/lec6 codes/1.py', wdir='C:/Users/Windows.10/Desktop/lec6 codes')
{'HTTP': 80, 'HTTPS': 433, 'FTP': 21, 'DNS': 53}
In [2]: runfile('C:/Users/Windows.10/Desktop/lec6 codes/3.py', wdir='C:/Users/Windows.10/Desktop/lec6 codes')
Bio
In [3]: runfile('C:/Users/Windows.10/Desktop/lec6 codes/4.py', wdir='C:/Users/Windows.10/Desktop/lec6 codes')
{0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}
In [4]:
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 1, Col 27
ASCII
CRLF
RW
Mem 64%
ENG

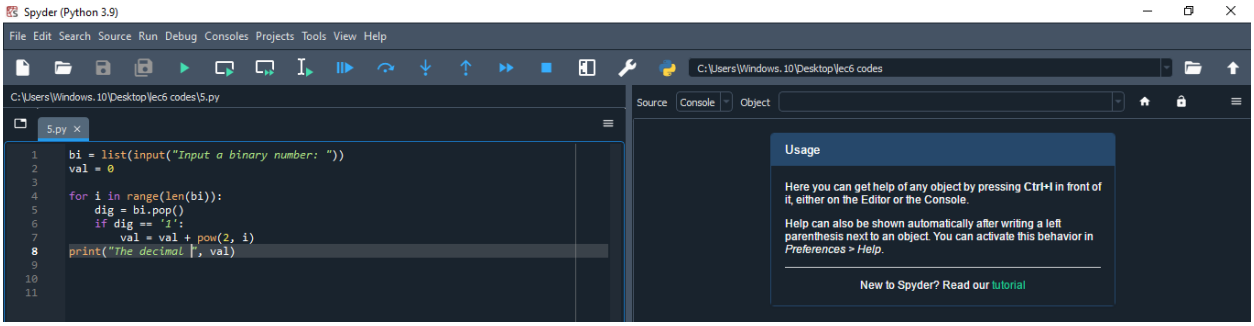
Question 2:

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.




```
Input a binary number: 10011
The decimal  19
```

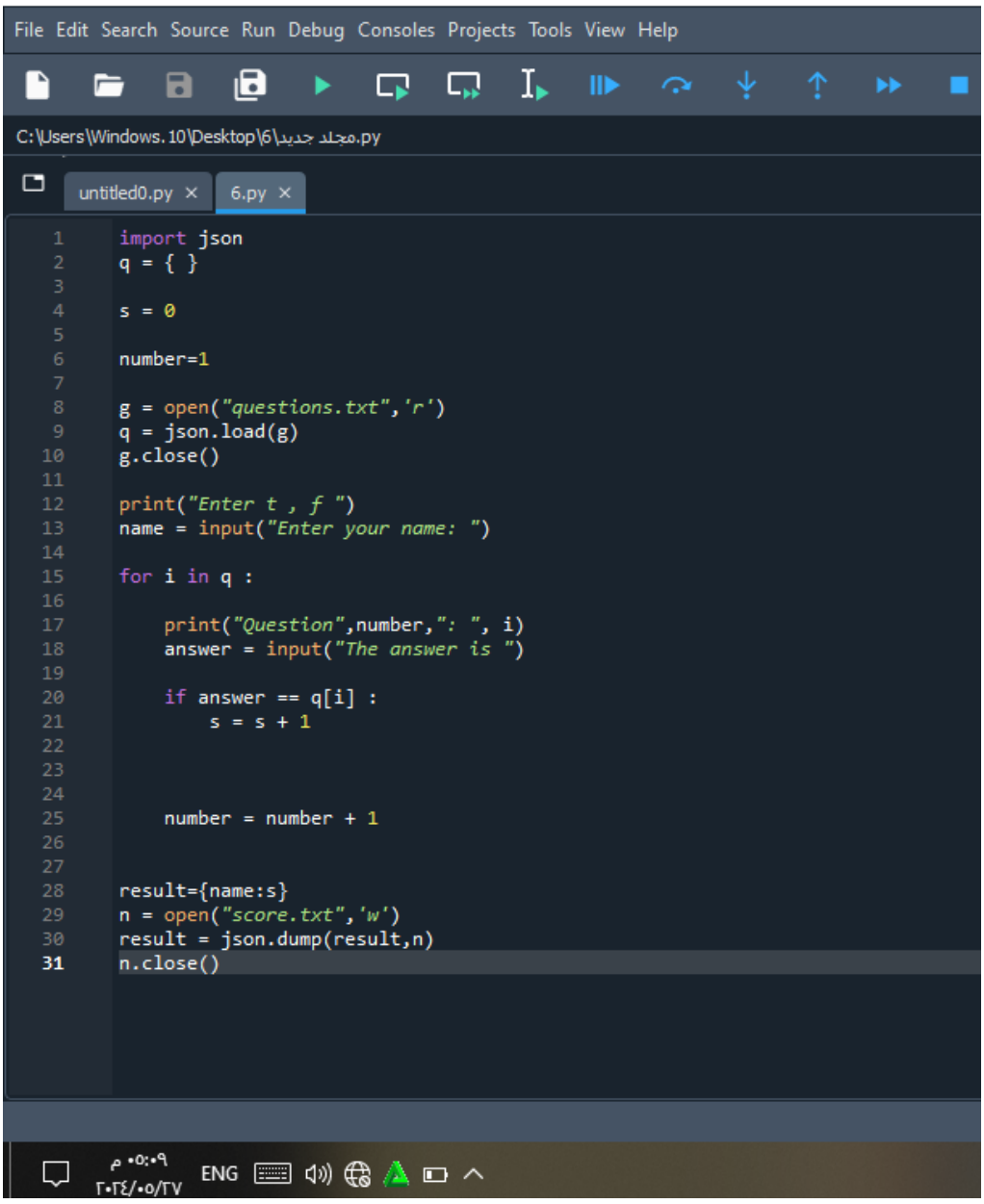
Question 3:

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

finally computes and prints user results and store user name and result in separate file csv or json file

Spyder (Python 3.9)

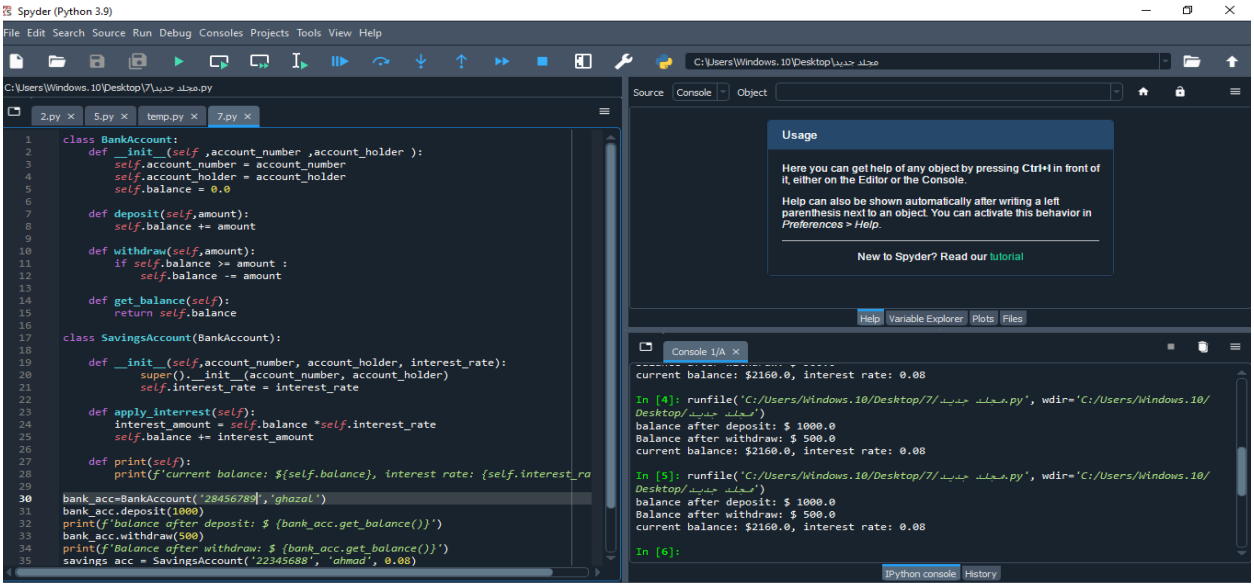


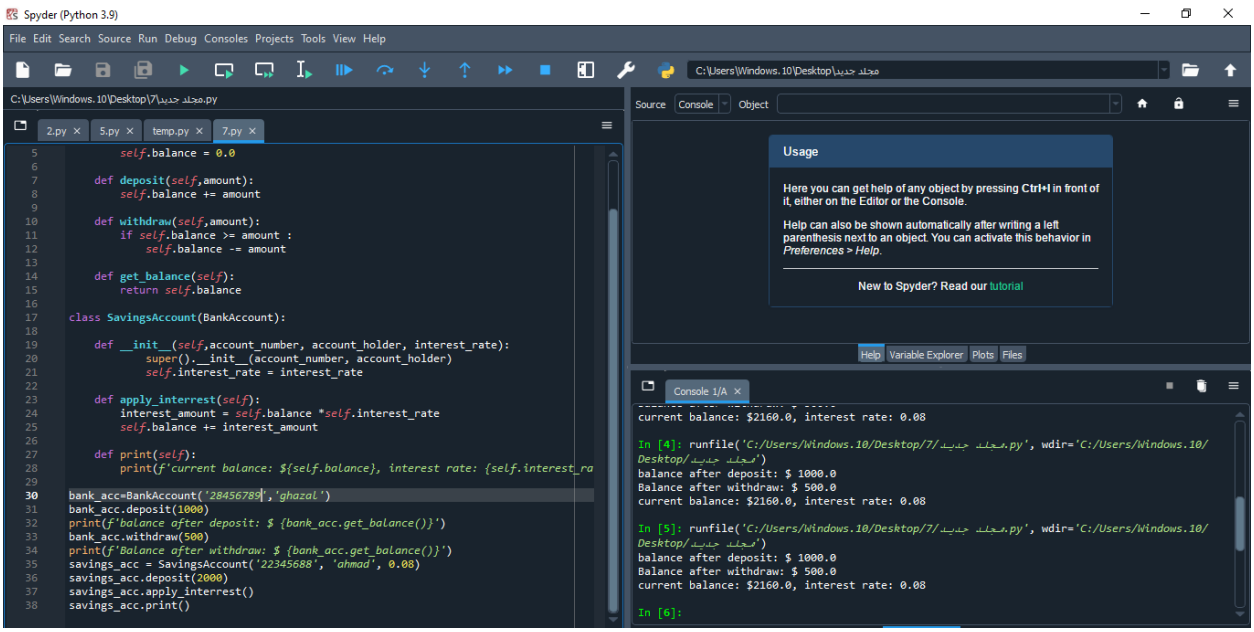
score - Notepad

File Edit Format View Help

{"GHazal ": 2}

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


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Windows.10\Desktop\مجلد جديد
C:\Users\Windows.10\Desktop\7\مجلد جديد.py
2.py
5.py
temp.py
7.py
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
def withdraw(self,amount):
if self.balance >= amount :
self.balance -= amount
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_interrest(self):
interest_amount = self.balance *self.interest_rate
self.balance += interest_amount
def print(self):
print(f'current balance: ${self.balance}, interest rate: {self.interest_ra
bank_acc=BankAccount('28456789','ghazal')
bank_acc.deposit(1000)
print(f'balance after deposit: $ {bank_acc.get_balance()}')
bank_acc.withdraw(500)
print(f'Balance after withdraw: $ {bank_acc.get_balance()}')
savings_acc = SavingsAccount('22345688', 'ahmad', 0.08)
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Help Variable Explorer Plots Files
Console 1/A
current balance: $2160.0, interest rate: 0.08
In [4]: runfile('C:/Users/Windows.10/Desktop/7/مجلد جديد.py', wdir='C:/Users/Windows.10/Desktop/مجلد جديد')
balance after deposit: $ 1000.0
Balance after withdraw: $ 500.0
current balance: $2160.0, interest rate: 0.08
In [5]: runfile('C:/Users/Windows.10/Desktop/7/مجلد جديد.py', wdir='C:/Users/Windows.10/Desktop/مجلد جديد')
balance after deposit: $ 1000.0
Balance after withdraw: $ 500.0
current balance: $2160.0, interest rate: 0.08
In [6]:
IPython console History


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Windows.10\Desktop\مجلد جديد
C:\Users\Windows.10\Desktop\7\مجلد جديد.py
2.py
5.py
temp.py
7.py
self.balance = 0.0
def deposit(self,amount):
self.balance += amount
def withdraw(self,amount):
if self.balance >= amount :
self.balance -= amount
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_interrest(self):
interest_amount = self.balance *self.interest_rate
self.balance += interest_amount
def print(self):
print(f'current balance: ${self.balance}, interest rate: {self.interest_ra
bank_acc=BankAccount('28456789','ghazal')
bank_acc.deposit(1000)
print(f'balance after deposit: $ {bank_acc.get_balance()}')
bank_acc.withdraw(500)
print(f'Balance after withdraw: $ {bank_acc.get_balance()}')
savings_acc = SavingsAccount('22345688', 'ahmad', 0.08)
savings_acc.deposit(2000)
savings_acc.apply_interrest()
savings_acc.print()
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Help Variable Explorer Plots Files
Console 1/A
current balance: $2160.0, interest rate: 0.08
In [4]: runfile('C:/Users/Windows.10/Desktop/7/مجلد جديد.py', wdir='C:/Users/Windows.10/Desktop/مجلد جديد')
balance after deposit: $ 1000.0
Balance after withdraw: $ 500.0
current balance: $2160.0, interest rate: 0.08
In [5]: runfile('C:/Users/Windows.10/Desktop/7/مجلد جديد.py', wdir='C:/Users/Windows.10/Desktop/مجلد جديد')
balance after deposit: $ 1000.0
Balance after withdraw: $ 500.0
current balance: $2160.0, interest rate: 0.08
In [6]: