

Name: جميل سليمان طوشان

Number: 1393

Homework Number: Homework 1

---

### Question 1: Python Basics

#### A. توليد متغير قاموس من قائمتين

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

```
for item in L:
    if item.startswith('B'):
        print(item)
```

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

Process finished with exit code 0

#### B. حساب العاملية

```
def factorial(n):
    if n == 0:
        return 1
    elif n < 0:
        return "Undefined"
    else:
        a = 1
        while n > 0:
            a = a * n
            n = n - 1
        return a
```

```
number = int(input("Enter a number: "))
result = factorial(number)
print(result)
```

Enter a number: 5

120

Process finished with exit code 0

### B تحديد العناصر التي تبدأ ب C.

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

for i in L:
    if i.startswith('B'):
        print(i)
```

Bio

Process finished with exit code 0

### D. Dictionary Comprehension

```
d={x:x+1 for x in range(0,11)}
print(d)
```

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

Process finished with exit code 0

### Question 2: Convert from Binary to Decimal

```
def func(bin):
    try:
        d = int(bin, 2)
        print(f"الثنائي للرقم العشري المكافئ {bin} هو {d}")
    except ValueError:
        print("صالح غير ثنائي رقم إدخال تم.")

s = input("ثنائي رقم أدخل")
func(s)
```

أدخل رقم ثنائي 1011

المكافئ العشري للرقم الثنائي 1011 هو 11

Process finished with exit code 0

### Question 3: Working with Files - Quiz Program

```
def extractfield(filename,n):
    infile=open(filename,"r")
    return [line.rstrip().split(',')[n-1] for line in infile]
score=0
infile=open("questions.csv",'r')
outfile=open("marks.csv","w")
questions=extractfield("questions.csv",1)
answers=extractfield("questions.csv",2)
for i in range(len(questions)):
    print(questions[i])
    answer=input("اجابتك:")
    if answer==answers[i]:
        score+=1
name=input("enter your name: ")
count=str(score)
print(name,count)
outfile.write(name+', '+count)
outfile.close()
```

نجهز ملف csv يحتوي على الأسئلة والإجابات بالشكل التالي:

الأسئلة والاجوبة

1+1,2  
2+2,4  
3+3,6  
4+4,8  
5+5,10  
6+6,12  
10\*10,100  
11\*11,121  
9\*9,81  
10\*10,100  
11\*11,121  
12\*12,144  
9\*9,81  
8\*8,64  
7-7,0  
6-6,0  
5-5,0  
3-3,0  
2-2,0  
1-1,0

الخرج:

$1+1$

اجابتك:2

$2+2$

اجابتك:4

$3+3$

اجابتك:6

$4+4$

اجابتك:8

$5+5$

اجابتك:10

$6+6$

اجابتك:12

$10*10$

اجابتك:100

$11*11$

اجابتك:121

$9*9$

اجابتك:81

$10*10$

اجابتك:100

$11*11$

اجابتك:121

$12*12$

اجابتك:144

$9*9$

اجابتك:81

$8*8$

اجابتك:2

7-7

اجابتك:0

6-6

اجابتك:0

5-5

اجابتك:0

3-3

اجابتك:0

2-2

اجابتك:0

1-1

اجابتك:0

enter your name: جميل

جميل 19

Process finished with exit code 0

ملف النتيجة:

	A	B	C
1	جميل	19	
2			
3			

#### Question 4: Object-Oriented Programming - Bank Class

```
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):
        if amount > 0:
            self.balance += amount
            print(f"الحالي الرصيد : ${self.balance}.")
        else:
            print("صفر من أكبر يكون أن يجب المبلغ.")

    def withdraw(self, amount):
        if 0 < amount <= self.balance:
            self.balance -= amount
            print(f"الحالي الرصيد : ${self.balance}.")
        else:
            print("كافي غير أو صالح غير المبلغ.")

    def get_balance(self):
        return self.balance

# العمليات وإجراء ميثيل إنشاء
account = BankAccount("1234", "جميل")
account.deposit(1000)
account.withdraw(500)
print(f"الحالي الرصيد : ${account.get_balance()}")

class SavingsAccount(BankAccount):
    # الكلاس هذه (Class) SavingsAccount ثرت (Inherits) من كلاس BankAccount
    def __init__(self, account_number, account_holder, balance=0.0,
interest_rate=0.0):
        # SavingsAccount بكلاس الخاص (Constructor) في إضافية قيم تعيين يتم
        super().__init__(account_number, account_holder, balance)
        self.interest_rate = interest_rate

    def apply_interest(self):
        # الرصيد على الفائدة تطبيق (Method) تابع هو هذا
        interest_amount = self.balance * self.interest_rate
        self.balance += interest_amount
        print(f"الحالي الرصيد .الفائدة تطبيق تم")

    def __str__(self):
        # والمعدل الرصيد يطبع الذي المُغَلَّل الطباعة (Method) تابع هو هذا
        return f"الحالي الرصيد : ${self.balance} ، الفائدة معدل : {self.interest_rate * 100}%"
```

```
# العمليات وإجراء (Instance) كائن إنشاء
savings_account = SavingsAccount("5678", "أحمد", 1000, 0.05)
savings_account.apply_interest()
print(savings_account)
```

في هذا الكود، قمنا بتعريف كلاسين BankAccount و SavingsAccount.

- كلاس BankAccount له ثلاثة خصائص account\_number ، account\_holder ، و balance. ولديه ثلاثة توابع: deposit() ، withdraw() ، و get\_balance().
- كلاس SavingsAccount يرث (Inherits) من كلاس BankAccount وله خاصية إضافية هي interest\_rate. كما أنه يحتوي على تابع apply\_interest() الذي يطبق الفائدة على الرصيد، وتابع \_\_str\_\_() المُعَدَّل الذي يطبع الرصيد والمعدل.

الخرج:

```
تم إيداع 1000$. الرصيد الحالي: 1000.0$
تم سحب 500$. الرصيد الحالي: 500.0$
الرصيد الحالي: 500.0$
تم تطبيق الفائدة. الرصيد الحالي: 1050.0$
الرصيد الحالي: 1050.0$ ، معدل الفائدة: 5.0%
```

Process finished with exit code 0