Network_Programming_Homework No.1

Name:

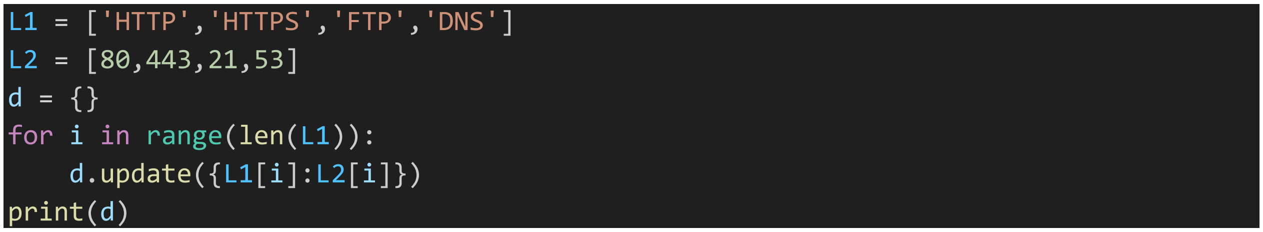
عمار يحيى علي

Num:

2564

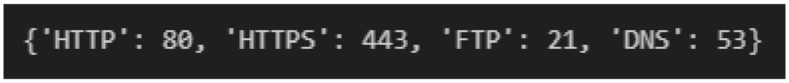
Question 1: Python Basics?

A- If you have to 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}



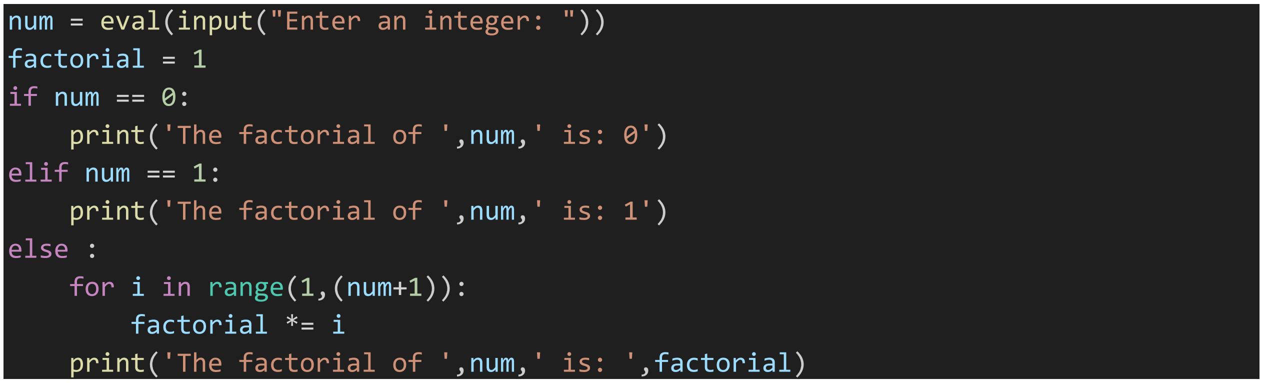
```
L1 = ['HTTP','HTTPS','FTP','DNS']
L2 = [80,443,21,53]
d = {}
for i in range(len(L1)):
    d.update({L1[i]:L2[i]})
print(d)
```

The output here is:



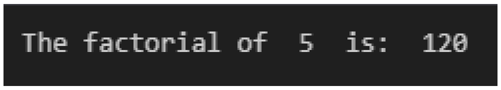
```
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
```

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



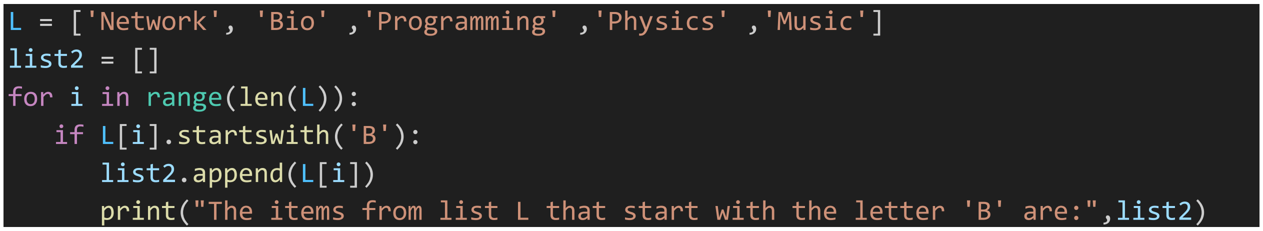
```
num = eval(input("Enter an integer: "))
factorial = 1
if num == 0:
    print('The factorial of ',num,' is: 0')
elif num == 1:
    print('The factorial of ',num,' is: 1')
else :
    for i in range(1,(num+1)):
        factorial *= i
    print('The factorial of ',num,' is: ',factorial)
```

The output here for in input = 5 is:



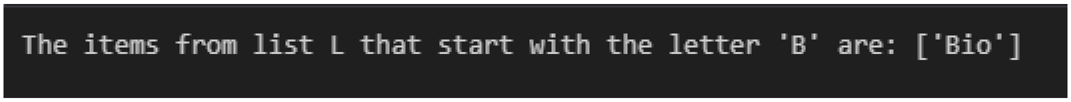
```
The factorial of  5  is:  120
```

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 the items that stars with ‘B’ letter, the print it on screen.



```
L = ['Network', 'Bio' ,'Programming' ,'Physics' ,'Music']
list2 = []
for i in range(len(L)):
   if L[i].startswith('B'):
      list2.append(L[i])
      print("The items from list L that start with the letter 'B' are:",list2)
```

The output here is:



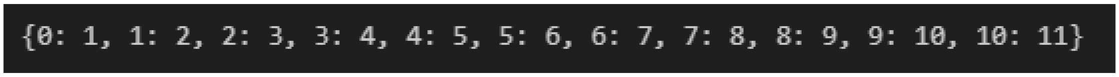
```
The items from list L that start with the letter 'B' are: ['Bio']
```

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

The output here is:



```
{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.

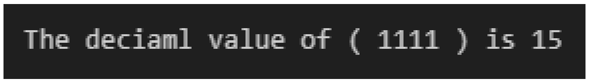
Write a Python program that converts a Binary number into its equivalent Decimal number.

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.

```
#defined function that calculates the decimal value of the enetered binary number.
def calculateDecimal(v1):
    sum = 0
    for i in range(len(v1)):
        sum += int(v1[i]) * (2**(len(v1)-1-i))
    print('The deciaml value of (',v1,') is',sum)
#my main code that allows the user to input and ensures the input is legit before passing it to the defined function.
binaryInput = input("Enter a binary number: ")
check = True
for i in range(len(binaryInput)):
    if binaryInput[i] == '1' or binaryInput[i] == '0':
        continue
    else :
        check = False
if check == True :
    calculateDecimal(binaryInput)
else :
    print('The number you entered (',binaryInput,') is not acceptable, Please rerun with a binary number consisted only of ones and zeroes.')
```

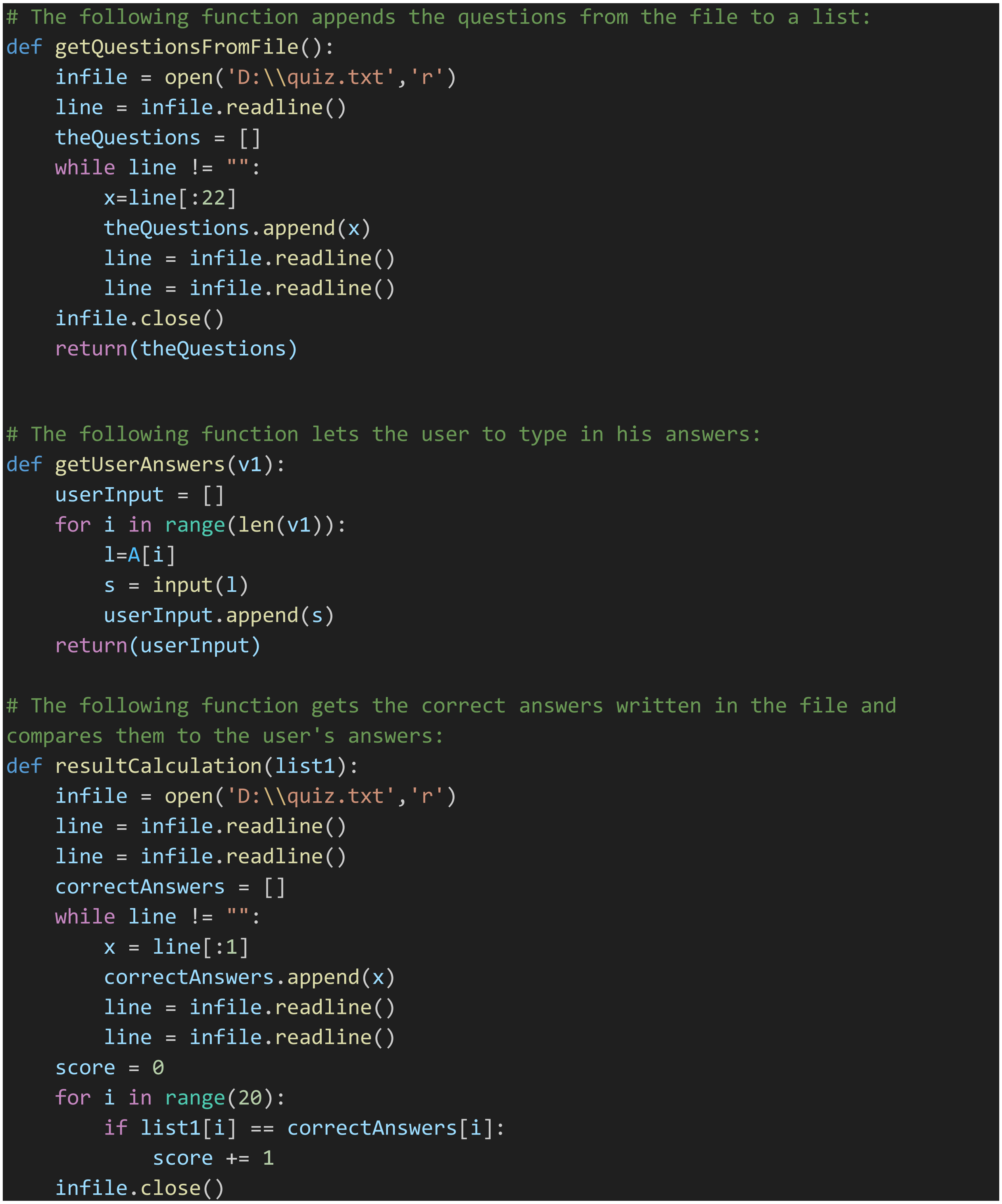
The output here for an input = 1111 is:



```
The deciaml value of ( 1111 ) is 15
```

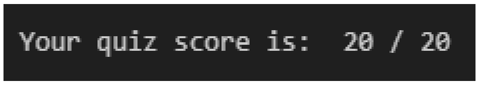
Question 3: Working with Files” Quiz Program”.

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 points user results and store user name and result in separate file csv or json file.



```
# The following function appends the questions from the file to a list:
def getQuestionsFromFile():
    infile = open('D:\\quiz.txt','r')
    line = infile.readline()
    theQuestions = []
    while line != "":
        x=line[:22]
        theQuestions.append(x)
        line = infile.readline()
        line = infile.readline()
    infile.close()
    return(theQuestions)


# The following function lets the user to type in his answers:
def getUserAnswers(v1):
    userInput = []
    for i in range(len(v1)):
        l=A[i]
        s = input(l)
        userInput.append(s)
    return(userInput)

# The following function gets the correct answers written in the file and
compares them to the user's answers:
def resultCalculation(list1):
    infile = open('D:\\quiz.txt','r')
    line = infile.readline()
    line = infile.readline()
    correctAnswers = []
    while line != "":
        x = line[:1]
        correctAnswers.append(x)
        line = infile.readline()
        line = infile.readline()
    score = 0
    for i in range(20):
        if list1[i] == correctAnswers[i]:
            score += 1
    infile.close()
    return(score)

#My main code that enables the user to input his name to start the quiz then create a .csv file and stores the result with the user's name in it.
import csv
fullName = input('Please Enter Your Name, Then answer with only with capital letter T or F.')
A = getQuestionsFromFile()
B = getUserAnswers(A)
C = resultCalculation(B)
print('Your quiz score is: ',C,'/ 20')
with open('RESULTS.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow([fullName, C])
```

The output here is:



```
Your quiz score is:  20 / 20
```

Question 4: Object-Oriented Programming - Bank ClassDefine.

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()

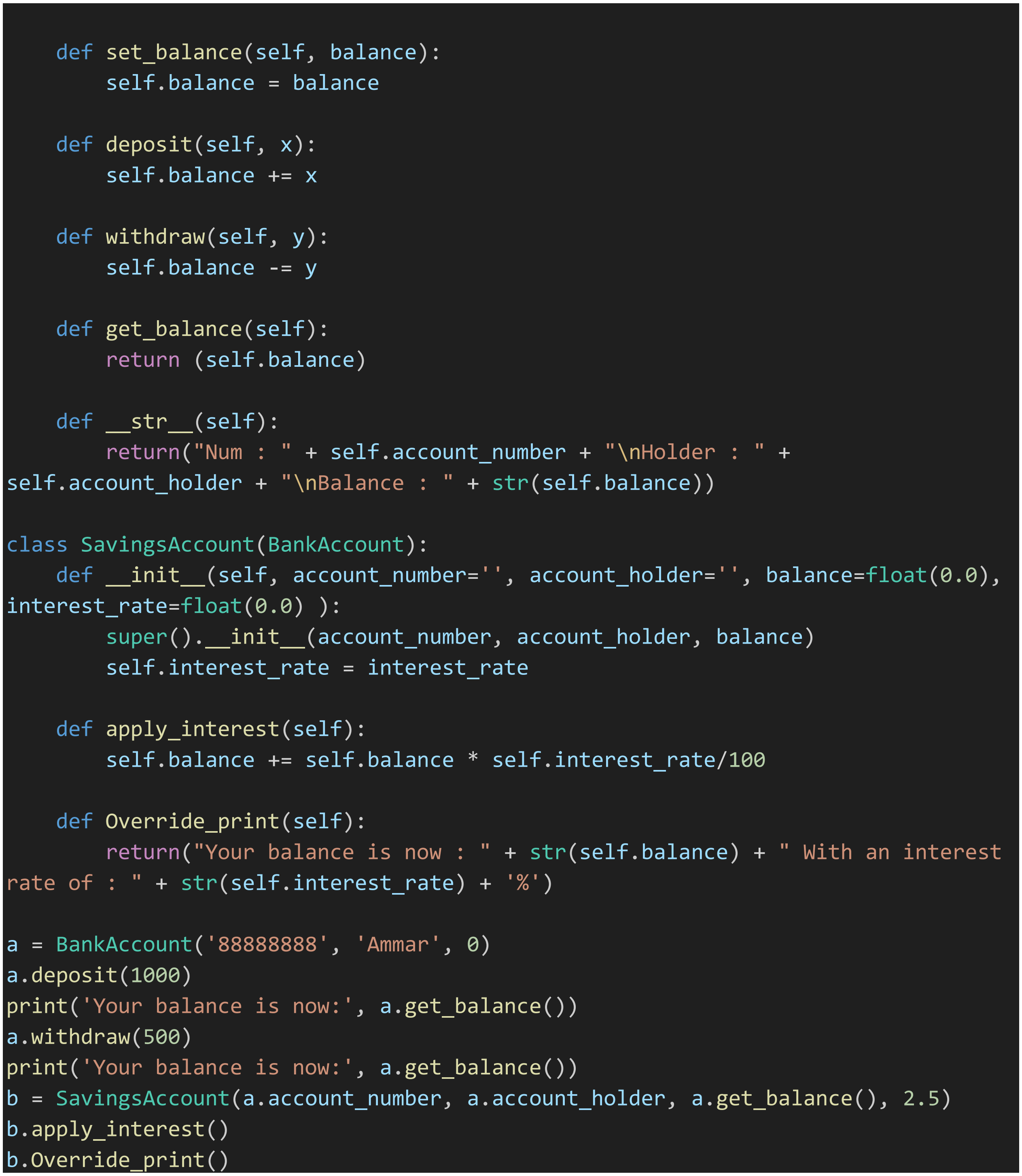
1- Create an instance of BankAccount, perform a deposit of $1000, Perform a withdrawal of $500.

2- Print the current balance after each operation.

3- 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.

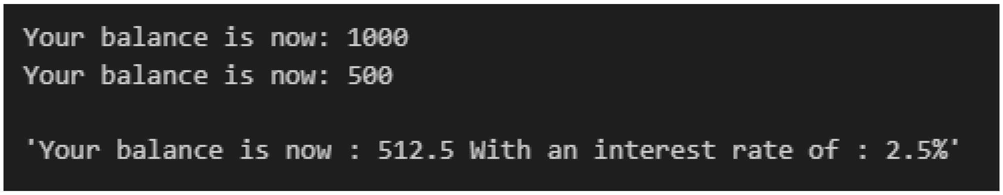
4- Create an instance of SavingsAccount, and call apply_interest() and print() functions.

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

    def set_account_number(self, account_number):
        self.account_number = account_number

    def set_account_holder(self, account_holder):
        self.account_holder = account_holder
```



```
    def set_balance(self, balance):
        self.balance = balance

    def deposit(self, x):
        self.balance += x

    def withdraw(self, y):
        self.balance -= y

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

    def __str__(self):
        return("Num : " + self.account_number + "\nHolder : " +
self.account_holder + "\nBalance : " + str(self.balance))

class SavingsAccount(BankAccount):
    def __init__(self, account_number='', account_holder='', balance=float(0.0),
interest_rate=float(0.0) ):
        super().__init__(account_number, account_holder, balance)
        self.interest_rate = interest_rate

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

    def Override_print(self):
        return("Your balance is now : " + str(self.balance) + " With an interest
rate of : " + str(self.interest_rate) + '%')

a = BankAccount('88888888', 'Ammar', 0)
a.deposit(1000)
print('Your balance is now:', a.get_balance())
a.withdraw(500)
print('Your balance is now:', a.get_balance())
b = SavingsAccount(a.account_number, a.account_holder, a.get_balance(), 2.5)
b.apply_interest()
b.Override_print()
```

The output here is:



```
Your balance is now: 1000
Your balance is now: 500

'Your balance is now : 512.5 With an interest rate of : 2.5%'
```