Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and electrical engineering

5th , Network Programming : Homework No1

الجمهورية العربية السورية

اللاذقية -جامعة تشرين

كلية الهندسة الكهربائية والميكانيكية

قسم هندسة الاتصالات والالكترونيات

السنة الخامسة: وظيفة1 برمجة شبكات

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 the 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 item that starts with 'B' letter, then print it on screen.

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}

Solution:

A- We can just use the zip function to convert it into a dictionary like this:

In [7]:
```
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- The idea here is to make the compiler calculate the factorial of the number entered like when we enter 1 the result would be 1, when we enter 5 the result would be 5!=5*4*3*2*1=120, and that's what this code does as shown below:

In [6]:
```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

def main():
    number = int(input("Enter a number: "))
    if number < 0:
        print("Factorial is not defined for negative numbers.")
    else:
        print(f"The factorial of {number} is {factorial(number)}")

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

```
Enter a number: 6
The factorial of 6 is 720
```

C- The code below is reading the items of the list then it uses the methods (startwith(), len(), loop())

```
In [8]: L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']

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

Bio
```

D- It need to write one instruction using the dictionary comprehension to do the task which is writing the numbers from 0 to 10 with the next number to each one of them as shown below.

```
In [9]: 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}
```

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.

Solution:

At first we define a function binary_to_decimal which take one variable which is the binary string that the user entered, then we create a loop that passes on every digit at the binary string and convert it using the equation
decimal_value=decimal_value*2+digit(binary)

Then we add the condition to avoid the invalid state when the user don't enter a binary number, and when he enter a number other than 0 or 1 like the shown below.

The first case: when the user enter a binary number and the program convert it into a decimal number:

In [11]:
```
def binary_to_decimal(binary_str):
    # Initialize decimal value
    decimal_value = 0

    # Convert the binary string to decimal
    for digit in binary_str:
        decimal_value = decimal_value * 2 + int(digit)

    return decimal_value

def main():
    # Read the binary number from the user
    binary_str = input("Enter a binary number: ")

    # Validate the input to ensure it's a binary number
    if not all(char in '01' for char in binary_str):
        print("Invalid binary number.")
        return

    # Calculate the decimal equivalent
    decimal_value = binary_to_decimal(binary_str)

    # Display the result
    print(f"The decimal equivalent of binary {binary_str} is {decimal_value}")

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

```
Enter a binary number: 010110111
The decimal equivalent of binary 010110111 is 183
```

The second case when the user enter a number other than 1 or 0:

In [12]:
```
def binary_to_decimal(binary_str):
    # Initialize decimal value
    decimal_value = 0

    # Convert the binary string to decimal
    for digit in binary_str:
        decimal_value = decimal_value * 2 + int(digit)

    return decimal_value

def main():
    # Read the binary number from the user
    binary_str = input("Enter a binary number: ")

    # Validate the input to ensure it's a binary number
    if not all(char in '01' for char in binary_str):
        print("Invalid binary number.")
        return

    # Calculate the decimal equivalent
    decimal_value = binary_to_decimal(binary_str)

    # Display the result
    print(f"The decimal equivalent of binary {binary_str} is {decimal_value}")

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

```
Enter a binary number: 324
Invalid binary number.
```

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.

Solution:

At first we need an instruction to read the file "question.json", we import the json and csv because we're gonna use them at the code later, then we define the functions: loading the questions from the json file to show it to the user and then enter the answers of the user and finally compare it with the right answers at the file that we choose at first, so the function reads a JSON file containing the quiz questions.

We open the file in read mode('r') , then the program reads the file and parses it as JSON , converting it into a Python list of dictionaries.

Then the questions appears to the user and the program checks the answers , questions are a list of dictionaries, the content of the JSON file is as shown below:

```
[
   {"question": "What is the capital of France?", "answer": "Paris"},
   {"question": "What is 2 + 2?", "answer": "4"},
   {"question": "What is the tallest mountain in the world?", "answer": "Mount Everest"}
]
```

So the function iterates through the questions using 'enumerate' to also get the index 'i'.

The nit checks if the answers is correct by comparing the user's answer after converting it to lower cases.

Then we return the count of the correct answers, and we save the results to a csv file after entering the user name and escort it with his score.

We can show a message to guide the user for each step and walk him through it all step by step.

At the end we show a message of saving the results of the quiz.

In [2]:
```
import json
import csv

def load_questions(filename):
    with open(filename, 'r') as file:
        questions = json.load(file)
    return questions

def ask_questions(questions):
    correct_answers = 0
    for i, q in enumerate(questions):
        print(f"Q{i + 1}: {q['question']}")
        answer = input("Your answer: ")
        if answer.strip().lower() == q['answer'].strip().lower():
            correct_answers += 1
    return correct_answers

def save_results(username, score, total_questions, results_file):
    with open(results_file, mode='a', newline='') as file:
        writer = csv.writer(file)
        writer.writerow([username, score, total_questions])

def main():
    questions_file = 'questions.json'
    results_file = 'results.csv'

    username = input("Enter your name: ")
    questions = load_questions(questions_file)
    total_questions = len(questions)

    print("\nStarting the quiz...\n")
    score = ask_questions(questions)

    print(f"\nQuiz completed! {username}, you scored {score} out of {total_questions}.\n")
```

```
    print(f"\nQuiz completed! {username}, you scored {score} out of {total_questions}.\n")

    save_results(username, score, total_questions, results_file)
    print("Your result has been saved.")

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

```
Enter your name: C:\Users\Soha\questions.json

Starting the quiz...

Q1: What is the capital of France?
Your answer: paris
Q2: What is 2 + 2?
Your answer: 4
Q3: What is the tallest mountain in the world?
Your answer: lol

Quiz completed! C:\Users\Soha\questions.json, you scored 2 out of 3.

Your result has been saved.
```

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.

Solution:

The code is very simple, it defines a BankAccount class taking attributes for the account , which is the account number, account holder and the balance with initial value of 0.0 then it defines a methods for depositing , withdrawing and getting the balance.

We use the inheriting rules to make new objects of the BankAccount Class by defining a SavingAccount subclass which adds to the parent class the attributes: interest rate and include a method to apply the interests to the balance. By overriding the print method in the parent class to show the balance and interest rate.

Creates an instance of BankAccount, performs a deposit and withdrawal, and prints the balance after each operation and Creates an instance of SavingsAccount, performs a deposit, applies interest, and prints the updated balance and interest rate.

The code is as shown below:

In [4]:
```
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):
        if amount > 0:
            self.balance += amount
            print(f"Deposited ${amount:.2f}. New balance is ${self.balance:.2f}")
        else:
            print("Deposit amount must be positive")

    def withdraw(self, amount):
        if 0 < amount <= self.balance:
            self.balance -= amount
            print(f"Withdrew ${amount:.2f}. New balance is ${self.balance:.2f}")
        else:
            print("Invalid withdrawal amount or insufficient funds")

    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 = self.balance * self.interest_rate / 100
        self.balance += interest
        print(f"Applied ${interest:.2f} interest. New balance is ${self.balance:.2f}")

    def print(self):
        print(f"Current balance: ${self.balance:.2f}, Interest rate: {self.interest_rate}%")
```

```
    def print(self):
        print(f"Current balance: ${self.balance:.2f}, Interest rate: {self.interest_rate}%")


# Create an instance of BankAccount
account = BankAccount("32465323", "Soha Darwish")

# Perform a deposit of $1000
account.deposit(1000)

# Perform a withdrawal of $500
account.withdraw(500)

# Print the current balance after each operation
print(f"Current balance: ${account.get_balance():.2f}")

# Create an instance of SavingsAccount
savings_account = SavingsAccount("32356423", "Soha Darwish", 5.0)

# Perform a deposit to SavingsAccount to apply interest on
savings_account.deposit(1000)

# Apply interest
savings_account.apply_interest()

# Print the current balance and interest rate
savings_account.print()
```

```
Deposited $1000.00. New balance is $1000.00
Withdrew $500.00. New balance is $500.00
Current balance: $500.00
Deposited $1000.00. New balance is $1000.00
Applied $50.00 interest. New balance is $1050.00
Current balance: $1050.00, Interest rate: 5.0%
```

End of Homework

Thanks for reading

Soha Yahia Darwish