**Assignment 20**

1. Set the variable test1 to the string 'This is a test of the emergency text system,' and save test1 to a file named test.txt.

A) test1 = 'This is a test of the emergency text system.'

# Save test1 to a file named test.txt

with open('test.txt', 'w') as file:

file.write(test1)

1. Read the contents of the file test.txt into the variable test2. Is there a difference between test 1 and test 2?

A) # Read the contents of test.txt into test2

with open('test.txt', 'r') as file:

test2 = file.read()

# Check if there's a difference between test1 and test2

print(test1 == test2)

This will print True if there is no difference between test1 and test2, indicating that the content read from the file is the same as the original string.

3. Create a CSV file called books.csv by using these lines:

title,author,year

The Weirdstone of Brisingamen,Alan Garner,1960

Perdido Street Station,China Miéville,2000

Thud!,Terry Pratchett,2005

The Spellman Files,Lisa Lutz,2007

Small Gods,Terry Pratchett,1992

A) csv\_content = """title,author,year

The Weirdstone of Brisingamen,Alan Garner,1960

Perdido Street Station,China Miéville,2000

Thud!,Terry Pratchett,2005

The Spellman Files,Lisa Lutz,2007

Small Gods,Terry Pratchett,1992

"""

# Write the CSV content to books.csv

with open('books.csv', 'w') as file:

file.write(csv\_content)

1. Use the sqlite3 module to create a SQLite database called books.db, and a table called books with these fields: title (text), author (text), and year (integer).

import sqlite3

# Connect to the SQLite database (creates it if not exists)

connection = sqlite3.connect('books.db')

# Create a cursor object to execute SQL queries

cursor = connection.cursor()

# Define the SQL query to create the books table

create\_table\_query = '''

CREATE TABLE IF NOT EXISTS books (

title TEXT,

author TEXT,

year INTEGER

)

'''

# Execute the SQL query to create the table

cursor.execute(create\_table\_query)

# Commit the changes and close the connection

connection.commit()

connection.close()

1. Read books.csv and insert its data into the book table.

A) import csv

import sqlite3

# Connect to the SQLite database

connection = sqlite3.connect('books.db')

# Create a cursor object to execute SQL queries

cursor = connection.cursor()

# Read data from books.csv and insert into the books table

with open('books.csv', 'r') as csvfile:

# Create a CSV reader

csv\_reader = csv.reader(csvfile)

# Skip the header row

next(csv\_reader, None)

# Iterate through the rows and insert into the books table

for row in csv\_reader:

title, author, year = row

cursor.execute('INSERT INTO books (title, author, year) VALUES (?, ?, ?)', (title, author, int(year)))

# Commit the changes and close the connection

connection.commit()

connection.close()

1. Select and print the title column from the book table in alphabetical order.

A) import sqlite3

# Connect to the SQLite database

connection = sqlite3.connect('books.db')

# Create a cursor object to execute SQL queries

cursor = connection.cursor()

# Select the title column from the books table in alphabetical order

cursor.execute('SELECT title FROM books ORDER BY title')

# Fetch all the results

titles = cursor.fetchall()

# Print the titles

for title in titles:

print(title[0])

# Close the connection

connection.close()

1. From the book table, select and print all columns in the order of publication.

A) import sqlite3

# Connect to the SQLite database

connection = sqlite3.connect('books.db')

# Create a cursor object to execute SQL queries

cursor = connection.cursor()

# Select all columns from the books table in the order of publication

cursor.execute('SELECT \* FROM books ORDER BY year')

# Fetch all the results

books = cursor.fetchall()

# Print the books

for book in books:

print(book)

# Close the connection

connection.close()

1. Use the sqlalchemy module to connect to the sqlite3 database books.db that you just made in exercise 6.

A) from sqlalchemy import create\_engine

# Create an SQLite database engine

engine = create\_engine('sqlite:///books.db')

# Connect to the database

connection = engine.connect()

# Now you can execute SQL queries using this connection

# Remember to close the connection when done

connection.close()

1. Install the Redis server and the Python redis library (pip install redis) on your computer. Create a Redis hash called test with the fields count (1) and name ('Fester Bestertester'). Print all the fields for test.

A) I don't have the ability to install software on your computer, but I can guide you through the process. To install the Redis server and the Python redis library, follow these steps:

Install Redis Server:

Visit the official Redis download page and download the appropriate version for your operating system.

Follow the installation instructions provided on the Redis website.

Install Python Redis Library:

Open your terminal or command prompt.

Run the following command to install the Redis library for Python using pip:

pip install redis

import redis

# Connect to the Redis server

redis\_client = redis.StrictRedis(host='localhost', port=6379, db=0)

# Create the 'test' hash

redis\_client.hset('test', 'count', 1)

redis\_client.hset('test', 'name', 'Fester Bestertester')

# Print all fields for 'test'

all\_fields = redis\_client.hgetall('test')

print(all\_fields)

1. Increment the count field of test and print it.

A) import redis

# Connect to the Redis server

redis\_client = redis.StrictRedis(host='localhost', port=6379, db=0)

# Increment the 'count' field of 'test'

redis\_client.hincrby('test', 'count', 1)

# Print the updated 'count' field

updated\_count = redis\_client.hget('test', 'count')

print(f"Updated count: {updated\_count}")