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.

ANSWER.

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

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

file.write(test1)

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

ANSWER.

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

test2 = file.read()

if test1 == test2:

print("There is no difference between test1 and test2.")

else:

print("There is a difference between test1 and test2.")

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

ANSWER.

labels = "title,author,year"

data = [

"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"

]

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

file.write(labels + '\n')

for line in data:

file.write(line + '\n')

4. 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).

ANSWER.

import sqlite3

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

cursor = conn.cursor()

create\_table\_query = '''

CREATE TABLE IF NOT EXISTS books (

title TEXT,

author TEXT,

year INTEGER

);

'''

cursor.execute(create\_table\_query)

conn.commit()

conn.close()

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

ANSWER.

import sqlite3

import pandas as pd

df = pd.read\_csv('books.csv')

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

df.to\_sql('books', conn, if\_exists='append', index=False)

conn.commit()

conn.close()

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

ANSWER.

import sqlite3

import pandas as pd

df = pd.read\_csv('books.csv')

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

df.to\_sql('books', conn, if\_exists='append', index=False)

conn.commit()

conn.close()

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

ANSWER.

import sqlite3

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

cursor = conn.cursor()

select\_query = '''

SELECT \*

FROM books

ORDER BY year;

'''

cursor.execute(select\_query)

rows = cursor.fetchall()

for row in rows:

print(row)

conn.close()

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

ANSWER.

from sqlalchemy import create\_engine

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

with engine.connect() as connection:

result = connection.execute("SELECT 'Connected successfully!'")

print(result.scalar())

9. 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.

ANSWER.

import redis

r = redis.Redis(host='localhost', port=6379, db=0)

r.hset('test', 'count', 1)

r.hset('test', 'name', 'Fester Bestertester')

print(r.hgetall('test'))

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

ANSWER.

import redis

r = redis.Redis(host='localhost', port=6379, db=0)

r.hincrby('test', 'count', 1)

print("Updated count:", r.hget('test', 'count').decode())