Data Persistence

1) Add Stock to Portfolio: Implement a method add_stock in the Portfolio class that takes a Stock object as a parameter and adds it to the portfolio.
2) Remove Stock from Portfolio: Implement a method remove_stock in the Portfolio class that takes a Stock object as a parameter and removes it from the portfolio.
3) Display Portfolio: Implement a method display_portfolio in the Portfolio class that displays the details of all the stocks in the portfolio.
4) Convert Portfolio to Dictionary: Implement a method to_dict in the Portfolio class that returns a dictionary representation of the portfolio. The dictionary should include a list of dictionaries representing each stock in the portfolio.
5) Save Portfolio to JSON File: Implement a method save_to_json in the Portfolio class that takes a file path as a parameter and saves the portfolio data to a JSON file. The file should contain the portfolio details in JSON format.
6) Save Portfolio to CSV File: Implement a method save_to_csv in the Portfolio class that takes a file path as a parameter and saves the portfolio data to a CSV file. The file should include the headers for symbol, name, and price, followed by the corresponding stock data.
7) Save Portfolio to SQLite Database: Implement a method save_to_database in the Portfolio class that takes a database file path as a parameter and saves the portfolio data to an SQLite database. Create a table named portfolio with columns for symbol, name, and price, and insert the stock data into the table.
8) Save Portfolio to Pickle File: Implement a method save_to_pickle in the Portfolio class that takes a file path as a parameter and saves the portfolio data to a pickle file using the pickle module.
9) Load Portfolio from JSON File: Implement a static method load_from_json in the Portfolio class that takes a JSON file path as a parameter and returns a new Portfolio instance with the stock data loaded from the file.
10) Load Portfolio from CSV File: Implement a static method load_from_csv in the Portfolio class that takes a CSV file path as a parameter and returns a new Portfolio instance with the stock data loaded from the file.
11) Load Portfolio from SQLite Database: Implement a static method load_from_database in the Portfolio class that takes a database file path as a parameter and returns a new Portfolio instance with the stock data loaded from the database.
12) Load Portfolio from Pickle File: Implement a static method load_from_pickle in the Portfolio class that takes a pickle file path as a parameter and returns a new Portfolio instance with the stock data loaded from the file using the pickle module.
13) Write sample code and proper test cases to test all the methods.