Class Exercise:

Iterator

You are working on a data analysis project that involves processing a large dataset containing information about different books in a library. Your task is to create a class that represents an iterator for this dataset. The iterator should allow you to iterate over the books based on specific criteria, such as filtering books by genre or sorting them by publication year.

To accomplish this, you need to implement a class called BookIterator with the following specifications:

1. The class should have a constructor that takes a list of books as a parameter and initializes an instance variable to store the books.
2. The class should have a method called \_\_iter\_\_() that returns the iterator object itself.
3. The class should have a method called \_\_next\_\_() that retrieves the next book in each iteration based on the specified criteria. If there are no more books, it should raise the StopIteration exception.
4. The class should have methods to set the criteria for filtering and sorting the books. These methods should allow the user to dynamically change the criteria during iteration.

Your task is to complete the implementation of the BookIterator class according to the given specifications. Once you have implemented the class, create an instance of it with a sample list of books, set different criteria for filtering and sorting, and iterate over the books, printing their titles.

class BookIterator:

    def \_\_init\_\_(self, books):

# Your code here

    def \_\_iter\_\_(self):

        return self

    def \_\_next\_\_(self):

# Your code here

    def filter\_books(self):

# Your code here

    def sort\_books(self, books):

# Your code here

    def set\_filter\_criteria(self, criteria):

# Your code here

    def set\_sort\_criteria(self, criteria):

# Your code here

# Sample list of books

books = [

    {"title": "Python Crash Course", "author": "Eric Matthes", "genre": "Programming", "year": 2015},

    {"title": "Clean Code", "author": "Robert C. Martin", "genre": "Programming", "year": 2008},

    {"title": "The Pragmatic Programmer", "author": "Andrew Hunt, David Thomas", "genre": "Programming", "year": 1999},

    {"title": "Design Patterns", "author": "Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides", "genre": "Programming", "year": 1994},

    {"title": "To Kill a Mockingbird", "author": "Harper Lee", "genre": "Fiction", "year": 1960},

    {"title": "1984", "author": "George Orwell", "genre": "Fiction", "year": 1949},

    {"title": "Pride and Prejudice", "author": "Jane Austen", "genre": "Fiction", "year": 1813}

]

# Create an instance of BookIterator

# Your code here

# Set filter criteria to only include programming books

# Your code here

# Iterate over the filtered books and print their titles

# Your code here

# Set sort criteria to sort books by publication year

# Your code here

# Iterate over all the books sorted by publication year and print their titles

# Your code here

Expected output:

Programming Books:

Python Crash Course

Clean Code

The Pragmatic Programmer

Design Patterns

Books Sorted by Year:

Pride and Prejudice

1984

To Kill a Mockingbird

Design Patterns

The Pragmatic Programmer

Clean Code

Python Crash Course