Lab Assignment: Collection Framework in Java

Objective:

Explore Java's Collection Framework by using `List`, `Set`, and `Map` interfaces and their common implementations. Understand how to manipulate and traverse collections effectively.

Requirements:

1. Create a program to perform the following:
   - Store a list of books using `ArrayList`. Each book should have attributes like `title`, `author`, and `price`.
   - Store and retrieve student marks using a `HashMap` with the student name as the key and their marks as the value.
   - Store unique student IDs using a `HashSet`.
2. Sort the `ArrayList` of books based on the price using `Comparator`.
3. Demonstrate traversing collections using both `forEach()` and `Iterator`.

Code Starter:

```
// Book.java
public class Book {
    private String title;
    private String author;
    private double price;

    // Constructor, getters, and setters
    public Book(String title, String author, double price) {
        this.title = title;
        this.author = author;
        this.price = price;
    }

    public String getTitle() { return title; }
    public double getPrice() { return price; }

    @Override
    public String toString() {
        return title + " by " + author + " - $" + price;
    }
}

// Main.java
import java.util.*;

public class Main {
    public static void main(String[] args) {
```

```
        List<Book> books = new ArrayList<>();
        books.add(new Book("Java Basics", "Author A", 29.99));
        books.add(new Book("Advanced Java", "Author B", 39.99));

        // TODO: Sort and display books
        Map<String, Integer> studentMarks = new HashMap<>();
        Set<Integer> studentIDs = new HashSet<>();

        // TODO: Add, retrieve, and manipulate data in collections
    }
}
```

Exercises:

1. Add functionality to update the marks of a specific student in the `HashMap`.
2. Implement a `TreeSet` to store sorted student IDs.
3. Add a search feature to find books by title.

Bonus Tasks:

1. Use a `LinkedHashMap` to maintain insertion order for student marks and print them in the same order.
2. Serialize and deserialize the list of books using `ObjectOutputStream` and `ObjectInputStream`.
3. Implement a custom exception `DuplicateBookException` to prevent adding a book with the same title.

Submission:
Submit your code and screenshots of your program's output.