

Programming I - Assignment 7

Student Name: Evelyn Toledo Lally

Student Number: 22102020

Question 1

Book Class

```
public class Book {

    //private variables
    private String title;
    private String author;

    //constructor
    public Book(String title, String author) {
        this.title=title;
        this.author= author;
    }

    //getters
    public String getTitle(){
        return this.title;
    }

    public String getAuthor(){
        return this.author;
    }

    //setters
    public void setTitle(String title){
        this.title=title;
    }

    public void setAuthor(String author){
        this.author=author;
    }

    //toString
    public String toString(){
        return "Title: " + this.title + "\n"+
            "Author: " + this.author + "\n\n";
    }
}
```

Library Class

```

import java.util.ArrayList;

public class Library {

    //private and protected variables
    private String name;
    protected ArrayList<Book> bookList= new ArrayList<>();

    //constructor
    public Library(String name){
        this.name = name;
    }

    //getters
    public String getName(){
        return this.name;
    }

    public ArrayList<Book> getBooks(){
        return this.bookList;
    }

    //method to add book objects to the arraylist
    public void addBook(Book book){
        bookList.add(book);
    }

    //method to remove book objects from the arraylist
    public void removeBook(Book book){
        bookList.remove(book);
    }

    //toString
    public String toString(){
        return "Name: " + name + "\n"
            + "Book List: " + bookList;
    }
}

```

Driver Class - Write/Read to File

```

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.List;

public class Driver {

    //main method

```

```

public static void main(String[] args) {

    //Create an instance of the Library object
    Library library = new Library("Main");

    //Populate the arrayList with a number of Books objects
    library.addBook(new Book("Title1", "Author1"));
    library.addBook(new Book("Title2", "Author2"));
    library.addBook(new Book("Title3", "Author3"));

    //calling the methods to write the library list content to a txt file
    writeToFile(library.getBooks());

    //calling the method to display the library file
    readFromFile("Library.txt");
}

//method to write the contents of the ArrayList to the text file Library.txt
public static void writeToFile(List<Book> list){
    try {

        FileWriter fileWriter = new FileWriter("Library.txt");

        BufferedWriter writer = new BufferedWriter(fileWriter);

        writer.write("Book List \n");
        for(Book element:list){
            writer.write(element.toString());
        }
        writer.close();

    } catch(IOException e){
        e.printStackTrace();
    }
}

//method to read the contents of the fileName
public static void readFromFile(String fileName){
    try{
        File library = new File(fileName);

        FileReader fileReader = new FileReader(library);

        BufferedReader reader = new BufferedReader(fileReader);

        String line = null;

        while ((line = reader.readLine()) != null){
            System.out.println(line);
        }
        reader.close();

    } catch (IOException e){
        e.printStackTrace();
    }
}

```

```
    }  
    }  
}
```

Console with output

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  
  
(base) evelyntoledo@evelyntoledo-mbp Assignment 7 % java Driver  
Book List  
Title: Title1  
Author: Author1  
  
Title: Title2  
Author: Author2  
  
Title: Title3  
Author: Author3  
  
(base) evelyntoledo@evelyntoledo-mbp Assignment 7 % █
```

Question 2

Book Class

```
import java.io.Serializable;  
  
//implemented the Serializable instance  
public class Book implements Serializable {  
  
    private String title;  
    private String author;  
  
    public Book(String title, String author) {  
        this.title=title;  
        this.author= author;  
    }  
  
    public String getTitle(){  
        return this.title;  
    }  
  
    public String getAuthor(){  
        return this.author;  
    }  
}
```

```

public void setTitle(String title){
    this.title=title;
}

public void setAuthor(String author){
    this.author=author;
}

public String toString(){
    return "Title: " + this.title + "\n"+
        "Author: " + this.author + "\n\n";
}
}

```

Library Class

```

import java.io.Serializable;
import java.util.ArrayList;

//implemented the Serializable instance
public class Library implements Serializable{

    private String name;
    protected ArrayList<Book> bookList= new ArrayList<>();

    public Library(String name){
        this.name = name;
    }

    public String getName(){
        return this.name;
    }

    public ArrayList<Book> getBooks(){
        return this.bookList;
    }

    public void addBook(Book book){
        bookList.add(book);
    }

    public void removeBook(Book book){
        bookList.remove(book);
    }

    public String toString(){
        return "Name: " + name + "\n"
            + "Book List: " + bookList;
    }
}

```

Driver Class - Serialize/Deserialize

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.List;

public class DriverS {

    public static void main(String[] args) {

        Library library = new Library("Main");

        library.addBook(new Book("Title1", "Author1"));
        library.addBook(new Book("Title2", "Author2"));
        library.addBook(new Book("Title3", "Author3"));

        //calling the method to serialize the list
        serialise(library.getBooks());

        //calling the method to display the deserialized list
        deserialise("Library.dat");

    }

    //method to selialize the list of books
    public static void serialise(List<Book> list) {
        try {
            // create a connection stream (write bytes)
            FileOutputStream fileStream = new FileOutputStream("Library.dat");

            // create a chain stream (allows objects to be written to a stream)
            ObjectOutputStream os = new ObjectOutputStream(fileStream);
            // call writeObject() on the Object stream
            os.writeObject(list);

            os.close();

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

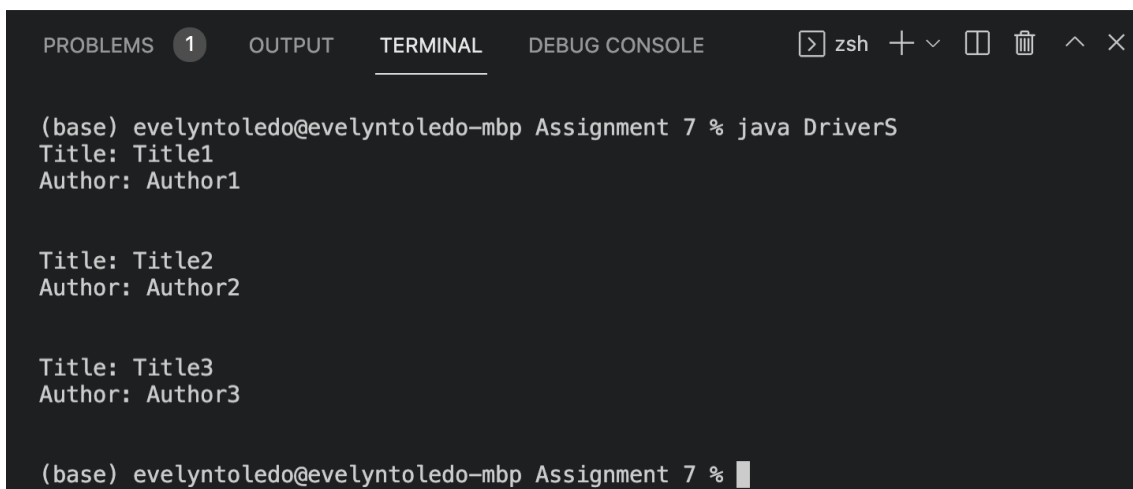
    //method to deserialize the list
    public static void deserialise(String fileName) {
        try{
            FileInputStream fileStream = new FileInputStream(fileName);

            ObjectInputStream os = new ObjectInputStream(fileStream);
```

```
        List<Book> books = (List<Book>)os.readObject();
        for (Book element:books){
            System.out.println(element.toString());
        }
        os.close();

    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
```

Console with output



The screenshot shows a terminal window with a dark background and light text. At the top, there is a tab bar with 'PROBLEMS' (containing a '1' icon), 'OUTPUT', 'TERMINAL' (which is active and underlined), and 'DEBUG CONSOLE'. To the right of the tabs are icons for a terminal window, a plus sign, a dropdown arrow, a window icon, a trash can, and up/down arrows. The terminal content shows the command `java Drivers` being executed, followed by three lines of output, each containing 'Title:' and 'Author:' on separate lines. The prompt at the bottom is `(base) evelyntoledo@evelyntoledo-mbp Assignment 7 %` followed by a cursor.

```
(base) evelyntoledo@evelyntoledo-mbp Assignment 7 % java Drivers
Title: Title1
Author: Author1

Title: Title2
Author: Author2

Title: Title3
Author: Author3

(base) evelyntoledo@evelyntoledo-mbp Assignment 7 %
```