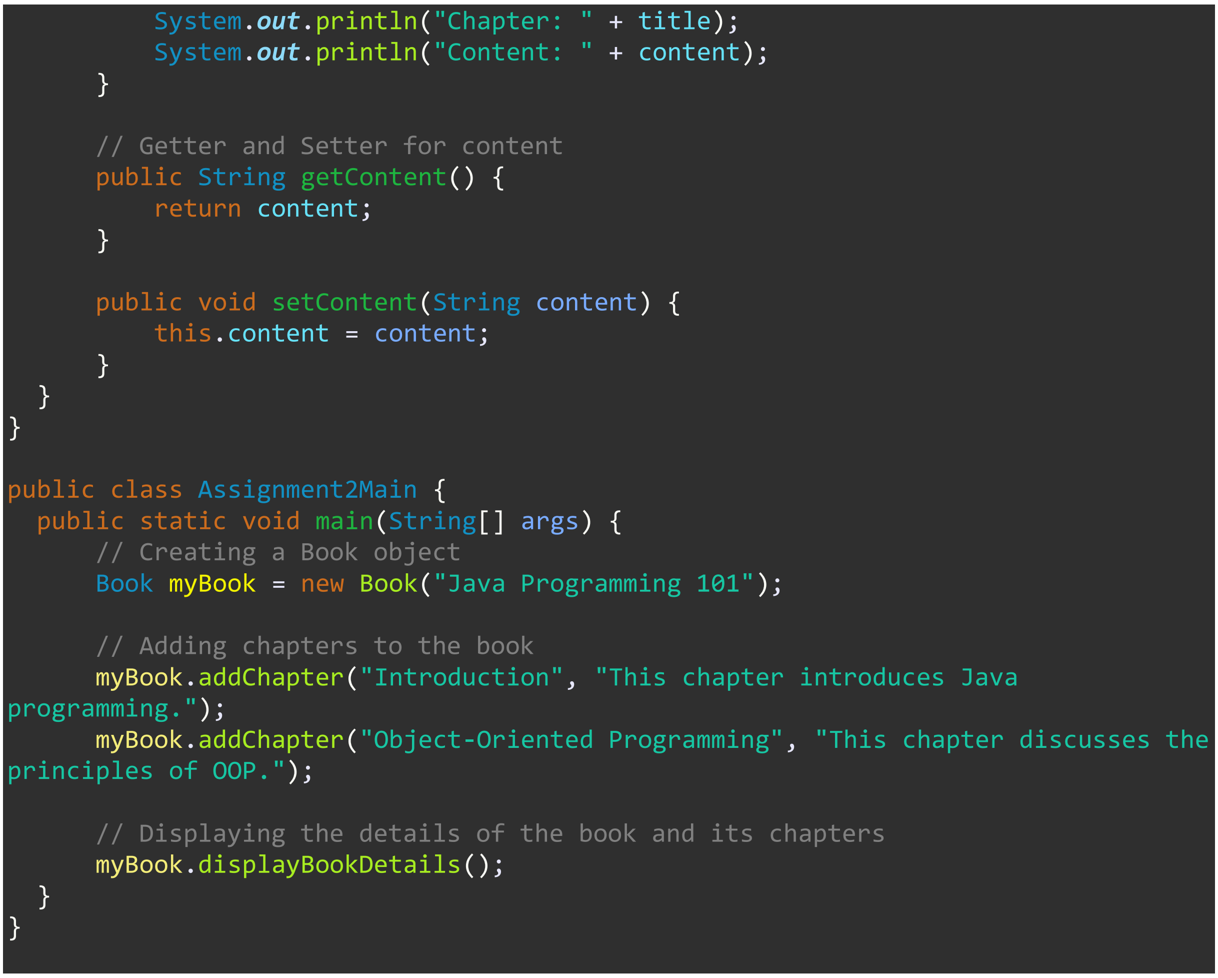
Module-8 Modular Programming

Problem statement-2: Explore the benefits and challenges of using nested classes. Implement a scenario where nested classes enhance code readability and maintainability

```
package modularProramming;

package modularProramming;

//Problem statement-2: Explore the benefits and challenges of using nested
classes.
//Implement a scenario where nested classes enhance code readability and
maintainability


import java.util.ArrayList;
import java.util.List;

//Outer class representing a Book
class Book {
  private String title;
  private List<Chapter> chapters;

  public Book(String title) {
      this.title = title;
      this.chapters = new ArrayList<>();
  }

  // Method to add chapters to the book
  public void addChapter(String title, String content) {
      chapters.add(new Chapter(title, content));
  }

  // Method to display book details
  public void displayBookDetails() {
      System.out.println("Book: " + title);
      for (Chapter chapter : chapters) {
          chapter.displayChapterDetails();
      }
  }

  // Nested class representing a Chapter
  class Chapter {
      private String title;
      private String content;

      public Chapter(String title, String content) {
          this.title = title;
          this.content = content;
      }

      // Method to display chapter details
      public void displayChapterDetails() {
```



```
          System.out.println("Chapter: " + title);
          System.out.println("Content: " + content);
      }

      // Getter and Setter for content
      public String getContent() {
          return content;
      }

      public void setContent(String content) {
          this.content = content;
      }
  }
}

public class Assignment2Main {
  public static void main(String[] args) {
      // Creating a Book object
      Book myBook = new Book("Java Programming 101");

      // Adding chapters to the book
      myBook.addChapter("Introduction", "This chapter introduces Java 
programming.");
      myBook.addChapter("Object-Oriented Programming", "This chapter discusses the 
principles of OOP.");

      // Displaying the details of the book and its chapters
      myBook.displayBookDetails();
  }
}
```