**Practical debugging with copilot**

**Problem statement**

The Library Management System you created earlier works well, but the following version of the code has some errors that need to be fixed. These errors prevent the program from correctly adding, removing, and displaying books. Use Microsoft Copilot to help debug the program.

**Starting Code with Errors**

**class** **LibraryManager**

{

**static** **void** **Main**()

{

**string** book1 = "";

**string** book2 = "";

**string** book3 = "";

**string** book4 = "";

**string** book5 = "";

**while** (**true**)

{

Console.WriteLine("Would you like to add or remove a book? (add/remove/exit)");

**string** action = Console.ReadLine();

**if** (action == "add")

{

Console.WriteLine("Enter the title of the book to add:");

**string** newBook = Console.ReadLine();

**if** (**string**.IsNullOrEmpty(book1))

{

book1 = newBook;

}

**else** **if** (**string**.IsNullOrEmpty(book2))

{

book2 = newBook;

}

**else** **if** (**string**.IsNullOrEmpty(book3))

{

book3 = newBook;

}

**else** **if** (**string**.IsNullOrEmpty(book4))

{

book4 = newBook;

}

**else** **if** (**string**.IsNullOrEmpty(book5))

{

book5 = newBook;

}

**else**

{

Console.WriteLine("The library is full. No more books can be added.");

}

}

**else** **if** (action == "remove")

{

Console.WriteLine("Enter the title of the book to remove:");

**string** removeBook = Console.ReadLine();

**if** (removeBook == book1)

{

book1 = "";

}

**else** **if** (removeBook == book2)

{

book2 = "";

}

**else** **if** (removeBook == book3)

{

book3 = "";

}

**else** **if** (removeBook == book4)

{

book4 = "";

}

**else** **if** (removeBook == book5)

{

book5 = "";

}

**else**

{

Console.WriteLine("Book not found.");

}

}

**else** **if** (action == "exit")

{

**break**;

}

**else**

{

Console.WriteLine("Invalid action. Please type 'add', 'remove', or 'exit'.");

}

// Display the list of books

Console.WriteLine("Available books:");

Console.WriteLine(book1);

Console.WriteLine(book2);

Console.WriteLine(book3);

Console.WriteLine(book4);

Console.WriteLine(book5);

}

}

}

**Steps to Complete the Task**

1. **Identify the Errors**Use the Visual Studio Code console application you created at the start of the course. Remove any existing code in the Program.cs file of your console application and create all the code in each step in that file. Run the **Starting Code with Errors** code and observe the issues:
   * Adding books without checking if the library is full first. This can lead to books being added even when the library is full, causing confusion.
   * Not checking for null or empty strings when displaying books: This can result in empty lines being printed, which can be confusing for the user.
   * The action variable comparison will only work if the user types a string with all lowercase letters. For example, if the user types “Add” the program will interpret this as an invalid command.
2. **Use Copilot to Debug**
   * Use Microsoft Copilot to identify and suggest fixes for the errors.
   * Apply the necessary corrections to ensure the program works as intended.
3. **Test the Debugged Program**
   * After making the necessary corrections, test the program by adding, removing, and displaying books.

**Code:**

**class** **LibraryManager**

{

**static** **void** **Main**()

{

**string**[] books = **new** **string**[**5**];

**while** (**true**)

{

Console.WriteLine("Would you like to add or remove a book? (add/remove/exit)");

**string** action = Console.ReadLine()?.Trim().ToLower();

**if** (action == "add")

{

**bool** hasSpace = **false**;

**foreach** (**var** book **in** books)

{

**if** (**string**.IsNullOrEmpty(book))

{

hasSpace = **true**;

**break**;

}

}

**if** (!hasSpace)

{

Console.WriteLine("The library is full. No more books can be added.");

**continue**;

}

Console.WriteLine("Enter the title of the book to add:");

**string** newBook = Console.ReadLine()?.Trim();

**for** (**int** i = **0**; i < books.Length; i++)

{

**if** (**string**.IsNullOrEmpty(books[i]))

{

books[i] = newBook;

Console.WriteLine($"Book \"{newBook}\" added.");

**break**;

}

}

}

**else** **if** (action == "remove")

{

Console.WriteLine("Enter the title of the book to remove:");

**string** removeBook = Console.ReadLine()?.Trim();

**bool** found = **false**;

**for** (**int** i = **0**; i < books.Length; i++)

{

**if** (books[i] == removeBook)

{

books[i] = "";

Console.WriteLine($"Book \"{removeBook}\" removed.");

found = **true**;

**break**;

}

}

**if** (!found)

{

Console.WriteLine("Book not found.");

}

}

**else** **if** (action == "exit")

{

Console.WriteLine("Exiting program...");

**break**;

}

**else**

{

Console.WriteLine("Invalid action. Please type 'add', 'remove', or 'exit'.");

}

// Display the list of books

Console.WriteLine("\nAvailable books:");

**foreach** (**var** book **in** books)

{

**if** (!**string**.IsNullOrEmpty(book))

{

Console.WriteLine($"- {book}");

}

}

Console.WriteLine();

}

}

}