ASSIGNMENT 1

■ **Task 2:** Document how the safe navigation operator improves the robustness of your code.

What is the Safe Navigation Operator?

The safe navigation operator(?.) is a feature that allows us to access object properties without worrying about 'null' values causing ‘NullPointerExceptions’. It's written as ?. and helps in making our code more robust and easier to read.

How the Safe Navigation Operator Works

When we use ?., it checks if the left-hand side of the operator is 'null'. If it is, the expression returns 'null' without throwing an exception. If it's not 'null', then only it evaluates the right-hand side of the expression.

How does the safe navigation operator improve the robustness of your code?

The safe navigation operator in Apex helps improve the robustness of our code by preventing ‘NullPointerExceptions’ and reducing the need for extensive 'null' checking. If any part of the chain is 'null', the entire expression returns 'null' instead of throwing an exception. Fewer lines of code and less complexity mean easier maintenance and lower risk of introducing bugs.

Scenario: SOQL Optimization

○ The performance of your queries needs improvement.

■ **Task 1:** Analyze your SOQL queries using Salesforce tools and optimize them for better performance.

■ **Task 2:** Document the changes made and the performance improvements observed.

Salesforce Tools

1. **Salesforce Developer Console**: You can use the Query Editor in the Developer Console to analyze your SOQL queries. The 'Execution Plan' feature provides insights into the cost and performance of your query.
2. **Debug Logs**: Enable debug logs to capture detailed information about your SOQL queries and Apex code execution.

We can Optimize our Queries by following methods :-

- **Selective Filters**: Use filters to only get the data you need, especially with indexed fields.
- **Specify Fields**: Avoid using SELECT *. Instead, specify the fields you need.
- **Efficient Loops**: Don't query inside loops. Query once and store results in a collection.
- **Limit and Offset**: Use these to manage large sets of data.
- **Bulk Processing**: Use batch processes for large amounts of data.

Example :-

#1
Before : [SELECT Id,Name, (SELECT Id,Name,Session_Name__c,Email__c FROM Attendees__r) FROM Training_Session__c**]**

After : [SELECT Id,Name, (SELECT Id,Name,Session_Name__c,Email__c FROM Attendees__r) FROM Training_Session__c **Limit 1**]

#2
Before : [SELECT Id FROM Training_Session__c WHERE Name != null **]**

After : [SELECT Id FROM Training_Session__c **WHERE Name Like 'New%' OR Name Like 'Test%' Or Name Like 'User%']**