Session 9 – AADK Task Report

Conditionals, Logic, and Null Safety in Kotlin

1. Objective

To strengthen foundational Kotlin programming skills by implementing conditionals, logical operations, and null safety mechanisms to prevent application crashes and ensure stable program execution.

2. Concepts Covered

- Nullable and Non-Nullable Types
- Safe Call Operator (?.)
- Elvis Operator (?:)
- toIntOrNull() for Safe Type Conversion
- if-else and when Conditionals
- Logical Operators (&&, ||, !)
- Defensive Programming Practices

3. Implementation with Code

```kotlin
fun main() {

    print("Enter student name: ")
    val name: String? = readLine()

    print("Enter marks: ")
    val marksInput: String? = readLine()

    val marks = marksInput?.toIntOrNull()

    if (name.isNullOrEmpty()) {
        println("Invalid name entered.")
        return
    }

    if (marks == null) {
        println("Invalid marks entered.")
        return
    }

    val result = when {
        marks >= 90 -> "Grade A"
        marks >= 75 -> "Grade B"
        marks >= 50 -> "Grade C"
        else -> "Fail"
    }

    println("Student: $name")
    println("Marks: $marks")
    println("Result: $result")
```

```
        }
```

4. Explanation

This program demonstrates safe handling of nullable input values using Kotlin's null safety features. The readLine() function returns nullable strings, which are safely processed using the safe call operator and toIntOrNull() to prevent crashes due to invalid input. Conditional statements validate user input before proceeding. The when expression is used for clean and readable grading logic. Early return statements ensure invalid data does not propagate further into the program.

5. Conclusion

The AADK Task 9 successfully reinforces core Kotlin fundamentals including condition handling, logical expressions, and null safety mechanisms. These concepts are critical in real-world Android application development to prevent runtime crashes and build stable applications.