Session 8 Lab Report

Title: Console-Based Art Gallery Navigator in Kotlin

1. Objective

To develop a console-based art gallery navigator using Kotlin data classes and functions to move forward and backward through artworks using wrapping logic.

2. Source Code

```
data class Artwork(
    val title: String,
    val artist: String,
    val year: Int
)

fun main() {

    val gallery = listOf(
        Artwork("Starry Night", "Vincent van Gogh", 1889),
        Artwork("Mona Lisa", "Leonardo da Vinci", 1503),
        Artwork("The Persistence of Memory", "Salvador Dali", 1931),
        Artwork("The Scream", "Edvard Munch", 1893)
    )

    var currentIndex = 0

    while (true) {
        displayArtwork(gallery[currentIndex])

        println("\nEnter command (next, prev, exit): ")
        when (readLine()?.lowercase()) {
            "next" -> currentIndex = nextIndex(currentIndex, gallery.size)
            "prev" -> currentIndex = previousIndex(currentIndex, gallery.size)
            "exit" -> {
                println("Exiting gallery navigator.")
                break
            }
            else -> println("Invalid command. Try again.")
        }
    }
}

fun displayArtwork(artwork: Artwork) {
    println("\n===== Artwork =====")
    println("Title : ${artwork.title}")
    println("Artist: ${artwork.artist}")
    println("Year  : ${artwork.year}")
}

fun nextIndex(current: Int, size: Int): Int {
    return (current + 1) % size
}

fun previousIndex(current: Int, size: Int): Int {
    return if (current - 1 < 0) size - 1 else current - 1
}
```

3. Explanation

- A data class named Artwork is used to store title, artist, and year.
- A list stores multiple Artwork objects representing the gallery.
- currentIndex keeps track of the currently displayed artwork.
- nextIndex() uses modulo (%) to wrap from last item to first.
- previousIndex() handles reverse wrapping from first item to last.
- The when condition processes user commands and updates navigation accordingly.

4. Conclusion

This lab strengthens understanding of data classes, list management, conditional logic, and cyclic navigation. It forms the foundation for implementing galleries, carousels, and navigation-based applications in Android development.