

Program:

```
fun main() {  
    val task = Task("Class Started")  
    task.setTaskDescription("Join Class")  
    task.status()  
    val choice = 2;  
    when (choice) {  
        1 -> {  
            task.editTitle("Wait for 2 Min")  
            println("Title updated to: ${task.title}")  
        }  
        2 -> {  
            task.editDescription("Join With in Two Min")  
            println("Description updated to: ${task.description}")  
        }  
        else -> {  
            println("Invalid choice")  
        }  
    }  
}
```

```
class Task(  
    var title: String,  
    var description: String? = null,  
    var status: Boolean = false  
) {  
    fun setTaskDescription(description: String) {  
        this.description = description  
    }  
    fun status(){  
        this.status = true  
    }  
  
    fun editTitle(newTitle: String){  
        this.title = newTitle  
    }  
    fun editDescription(newDesc: String){  
        this.description = newDesc  
    }  
}
```

- Implement a task management system where each task has a title, description, and status.
- Use methods like setTaskDescription, editTitle, and editDescription to modify task details based on user interactions or system events

```

class Task(
    var id: Int?,
    var taskName: String?,
    var priority: Priority?
){
    init {
        try {
            requireNotNull(id) { "id must not be null" }
            requireNotNull(taskName) { "taskName must not be null" }
            requireNotNull(priority) { "priority must not be null" }
        } catch (e: IllegalArgumentException) {
            println("Error creating task: ${e.message}")
        }
    }

    override fun toString(): String {
        return "Todo(id=$id, taskName='$taskName', priority=$priority)"
    }
}

enum class Priority {
    HIGH, MEDIUM, LOW
}

fun main() {
    val tasks = listOf(
        Task(1, "Join Class", Priority.HIGH),
        Task(null, "Maintain Attendance", Priority.MEDIUM),
        Task(3, "Buy Books", null)
    )

    for (task in tasks) {
        println(task)
    }
}

```