### **1. Problem Statement:**

You are tasked with creating a task management system where tasks can be sorted by either priority or start time. Users should be able to switch between these sorting methods dynamically, allowing flexibility without changing the core system logic. This will be achieved using the Strategy Pattern.

### **2. Explanation of Code:**

1. Task Class:
   * Represents a task with a name, start time, and priority.
   * toString() method formats the task details for easy display.
2. SortStrategy Interface:
   * Defines the structure for sorting algorithms with a sort() method, ensuring that different sorting strategies can be applied to the task list.
3. SortByPriority Class:
   * Implements sorting based on priority in descending order, with higher priority tasks appearing first.
4. SortByStartTime Class:
   * Implements sorting by start time in ascending order, so tasks with earlier start times appear first.
5. TaskManager Class:
   * Manages tasks and allows for switching between sorting strategies using the setSortStrategy() method.
   * sortTasks() applies the chosen strategy, and displayTasks() prints the tasks.
6. Main Class (StrategyPatternExample):
   * Demonstrates the usage of the Strategy Pattern by adding tasks and switching between sorting by priority and sorting by start time.

### **3. Output**

