



H-TechServices

Assignment<4> - Fall 2024

Course Title:	Java	Course Code:	Java-01	Credit Hours:	4(3,1)
Course Instructor:	Muhammad Hasnat Rasool	Program Name:	Java Mastery		
Due Date:	010-12-2024	Maximum Marks:	10		

Important Instructions / Guidelines:

The submission date is **Dec 03, 2024**. Submit your assignment in the form of a report. It should contain a problem statement, solution (code), and output. Your pdf/docx file name should be your name.

Upload your file on **Github**.

Ensure your program runs without errors and follows the structure.

Learning Objectives: Java programming.

Problem Statement

You are tasked with designing a **Task Management System** for a company. This system will manage different types of tasks, each with varying priority levels and complexity. Employees can perform tasks, and each task may have custom completion rules depending on its type. Tasks should be categorized as **SimpleTask**, **ComplexTask**, or **UrgentTask**, each having its own priority calculation logic.

Requirements

1. Inheritance:

- Create a base abstract class Task with common attributes such as taskId, description, and priorityLevel.
- Subclass Task into three types: SimpleTask, ComplexTask, and UrgentTask. Each should override methods to define specific behaviors.

2. Polymorphism:

- Use polymorphism to manage tasks dynamically, allowing operations such as calculating priority and marking a task as complete to behave differently for each task type.

3. Abstract Classes and Interfaces:

- Use an interface `Taskable` with methods like `markComplete()` and `getPriority()`.
- Define `Task` as an abstract class to enforce implementation of shared behaviors by subclasses.

4. Composition:

- Use a `TaskManager` class to store and manage a collection of tasks using composition.
-

Class Requirements

Task (Abstract Class)

Attributes:

- `int taskId`
- `String description`
- `int priorityLevel`
- `boolean isCompleted`

Methods:

- `markComplete()` (abstract)
 - `getPriority()` (abstract)
 - `toString()` (common behavior to display task details)
-

SimpleTask (Subclass)

Attributes:

- Same as `Task`.

Methods:

- `getPriority()`: Priority is calculated as `priorityLevel * 1`.
 - `markComplete()`: Sets `isCompleted` to true.
-

ComplexTask (Subclass)

Attributes:

- Same as Task.

Methods:

- `getPriority()`: Priority is calculated as `priorityLevel * 2`.
 - `markComplete()`: Task is marked complete only if it is reviewed (use a condition `isReviewed`).
-

UrgentTask (Subclass)

Attributes:

- Same as Task.

Methods:

- `getPriority()`: Priority is calculated as `priorityLevel * 3`.
 - `markComplete()`: Adds a log message like "Urgent task completed."
-

Taskable (Interface)

Methods:

- `void markComplete()`
 - `int getPriority()`
-

TaskManager (Composition Class)

Attributes:

- `List<Task> tasks`

Methods:

- `void addTask(Task task)`: Adds a task to the list.

- `void completeTask(int taskId)`: Marks the task as complete using polymorphism.
 - `void showAllTasks()`: Displays all tasks in the list, sorted by priority.
-

Goal to Achieve

- Implement a system where the behavior of tasks depends on their type, ensuring that tasks with higher priority are handled differently.
 - Demonstrate polymorphism by working with tasks dynamically through the `Taskable` interface.
-

Sample Output

Task Added: Simple Task - Write Report

Task Added: Complex Task - Develop Module

Task Added: Urgent Task - Fix Server Issue

Tasks Sorted by Priority:

1. Urgent Task (Priority: 90) - Fix Server Issue
2. Complex Task (Priority: 40) - Develop Module
3. Simple Task (Priority: 10) - Write Report

Marking Task as Complete: Complex Task - Develop Module

Task cannot be completed: Complex Task requires review.

Marking Task as Complete: Urgent Task - Fix Server Issue

Task completed: Fix Server Issue (Urgent Task).

Marks Distribution (Total 10 Marks)

- 1. Abstract Class and Interface Implementation (2 Marks)**
- 2. Correct Implementation of Priority Calculation (3 Marks)**
- 3. Dynamic Behavior of markComplete () (Polymorphism) (3 Marks)**
- 4. Proper Sorting and Display of Tasks (2 Marks)**