

RIPHAH INTERNATIONAL UNIVERSITY GULBERG GREEN CAMPUS, ISLAMABAD.



Lab# 10

Course: SCD

Submitted by: Eman Fatima

Sap id:48821

BSSE -5 Semester

Submitted to: Ma'am Shazwa

Date of submission: 04 Nov, 2024

Department of BSSE

TASK 01:

Code:

```
// Model: Task Class
class Task {
    private int id;
    private String title;
    private String description;
    private boolean isCompleted;

    public Task(int id, String title, String description) {
        this.id = id;
        this.title = title;
        this.description = description;
        this.isCompleted = false;
    }

    // Getters
    public int getId() {
        return id;
    }

    public String getTitle() {
        return title;
    }

    public String getDescription() {
        return description;
    }

    public boolean isCompleted() {
        return isCompleted;
    }

    // Setters
    public void setTitle(String title) {
```

```

        this.title = title;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public void markAsCompleted() {
        this.isCompleted = true;
    }

    @Override
    public String toString() {
        return "ID: " + id + ", Title: " + title + ", Description: " +
description + ", Completed: " + isCompleted;
    }
}

// View: TaskView Class
class TaskView {
    public void displayTask(Task task) {
        System.out.println(task.toString());
    }

    public void displayTasks(java.util.List<Task> tasks) {
        if (tasks.isEmpty()) {
            System.out.println("No tasks available.");
        } else {
            for (Task task : tasks) {
                displayTask(task);
            }
        }
    }

    private java.util.Scanner scanner;

    public TaskView() {
        this.scanner = new java.util.Scanner(System.in);
    }

    public String promptForTitle() {
        System.out.print("Enter task title: ");
        return scanner.nextLine().trim();
    }
}

```

```

    public String promptForDescription() {
        System.out.print("Enter task description: ");
        return scanner.nextLine().trim();
    }

    public int promptForTaskId() {
        System.out.print("Enter task ID to mark as completed: ");
        while (true) {
            try {
                return scanner.nextInt();
            } catch (java.util.InputMismatchException e) {
                System.out.println("Invalid input. Please enter an integer.");
                scanner.next(); // clear invalid input
            }
        }
    }

    // finally block mein scanner close karo
    public void close() {
        scanner.close();
    }
}

// Controller: TaskController Class
class TaskController {
    private java.util.List<Task> tasks;
    private TaskView view;
    private int nextId;

    public TaskController(TaskView view) {
        this.tasks = new java.util.ArrayList<>();
        this.view = view;
        this.nextId = 1;
    }

    public void addTask(String title, String description) {
        if (tasks.stream()
            .anyMatch(task -> task.getTitle().equals(title) &&
task.getDescription().equals(description))) {
            System.out.println("Task already exists.");
        } else {
            Task task = new Task(nextId++, title, description);
            tasks.add(task);
        }
    }
}

```

```

        System.out.println("Task added.");
    }
}

public void displayTasks() {
    view.displayTasks(tasks);
}

public void markTaskAsCompleted(int id) {
    for (Task task : tasks) {
        if (task.getId() == id) {
            task.markAsCompleted();
            System.out.println("Task marked as completed.");
            return;
        }
    }
    System.out.println("Task not found.");
}
}

// Main Application
public class Main {
    public static void main(String[] args) {
        TaskView view = new TaskView();
        TaskController controller = new TaskController(view);
        java.util.Scanner scanner = new java.util.Scanner(System.in);

        try {
            while (true) {
                System.out.println("\n1. Add Task\n2. View Tasks\n3. Complete Task\n4. Exit");
                System.out.print("Choose an option: ");
                int choice = scanner.nextInt();
                scanner.nextLine(); // consume newline

                switch (choice) {
                    case 1:
                        String title = view.promptForTitle();
                        String description = view.promptForDescription();
                        controller.addTask(title, description);
                        break;
                    case 2:
                        controller.displayTasks();
                        break;

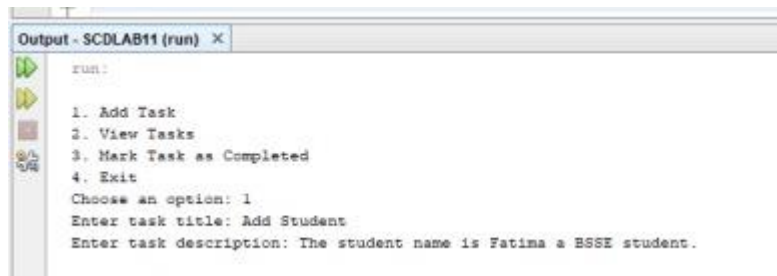
```

```

        case 3:
            int taskId = view.promptForTaskId();
            controller.markTaskAsCompleted(taskId);
            break;
        case 4:
            System.out.println("Exiting application.");
            scanner.close();
            return;
        default:
            System.out.println("Invalid choice. Please try again.");
    }
}
} catch (Exception e) {
    System.out.println("An error occurred: " + e.getMessage());
} finally {
    scanner.close();
}
}
}

```

OutPut:



```

1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Exit
Choose an option: 3
Enter task ID to mark as completed: 1
Task 1 marked as completed.

1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Exit
Choose an option: 3
Enter task ID to mark as completed: 2
Task 2 marked as completed.

1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Exit
Choose an option: 2
ID: 1, Title: Add Student, Description: The student name is Fatima a BSSE student., Status: Completed
ID: 2, Title: Add Student, Description: The student name is Hina a BSCE student., Status: Completed

1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Exit
Choose an option: 4
Exiting the application.
BUILD SUCCESSFUL (total time: 1 minute 34 seconds)

```