```
import java.io.*;
import java.util.*;

class Task {
    String title;
    String dueDate;
    boolean isCompleted;

    public Task(String title, String dueDate) {
        this.title = title;
        this.dueDate = dueDate;
        this.isCompleted = false;
    }

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

    @Override
    public String toString() {
        return (isCompleted ? "[Completed] " : "[Pending] ") + title +
(dueDate.isEmpty() ? "" : " (Due: " + dueDate + ")");
    }
}

public class ToDoListApp {
    private static final String FILE_NAME = "tasks.txt";
    private static List<Task> tasks = new ArrayList<>();

    public static void main(String[] args) {
        loadTasksFromFile();
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\n--- To-Do List Menu ---");
            System.out.println("1. Add Task");
            System.out.println("2. Remove Task");
            System.out.println("3. Mark Task as Completed");
            System.out.println("4. View Pending Tasks");
            System.out.println("5. View Completed Tasks");
            System.out.println("6. Exit");
            System.out.print("Choose an option: ");

            int choice = scanner.nextInt();
            scanner.nextLine();  // Consume newline

            switch (choice) {
                case 1:
                    addTask(scanner);
                    break;
```

```
            case 2:
                removeTask(scanner);
                break;
            case 3:
                markTaskCompleted(scanner);
                break;
            case 4:
                viewTasks(false);
                break;
            case 5:
                viewTasks(true);
                break;
            case 6:
                saveTasksToFile();
                System.out.println("Exiting... Tasks saved.");
                scanner.close();
                return;
            default:
                System.out.println("Invalid choice. Try again.");
        }
    }
}

private static void addTask(Scanner scanner) {
    System.out.print("Enter task title: ");
    String title = scanner.nextLine();
    System.out.print("Enter due date (or press Enter to skip): ");
    String dueDate = scanner.nextLine();
    tasks.add(new Task(title, dueDate));
    System.out.println("Task added successfully.");
}

private static void removeTask(Scanner scanner) {
    if (tasks.isEmpty()) {
        System.out.println("No tasks to remove.");
        return;
    }
    viewAllTasks();
    System.out.print("Enter task number to remove: ");
    int index = scanner.nextInt();
    if (index > 0 && index <= tasks.size()) {
        tasks.remove(index - 1);
        System.out.println("Task removed successfully.");
    } else {
        System.out.println("Invalid task number.");
    }
}

private static void markTaskCompleted(Scanner scanner) {
    viewTasks(false);
```

```
        System.out.print("Enter task number to mark as completed: ");
        int index = scanner.nextInt();
        if (index > 0 && index <= tasks.size() && !tasks.get(index -
1).isCompleted) {
            tasks.get(index - 1).markCompleted();
            System.out.println("Task marked as completed.");
        } else {
            System.out.println("Invalid selection.");
        }
    }

    private static void viewTasks(boolean completed) {
        System.out.println("\n--- " + (completed ? "Completed" : "Pending") + "
Tasks ---");
        int count = 0;
        for (int i = 0; i < tasks.size(); i++) {
            if (tasks.get(i).isCompleted == completed) {
                System.out.println((count + 1) + ". " + tasks.get(i));
                count++;
            }
        }
        if (count == 0) {
            System.out.println("No tasks found.");
        }
    }

    private static void viewAllTasks() {
        System.out.println("\n--- All Tasks ---");
        for (int i = 0; i < tasks.size(); i++) {
            System.out.println((i + 1) + ". " + tasks.get(i));
        }
        if (tasks.isEmpty()) {
            System.out.println("No tasks found.");
        }
    }

    private static void saveTasksToFile() {
        try (PrintWriter writer = new PrintWriter(new FileWriter(FILE_NAME))) {
            for (Task task : tasks) {
                writer.println(task.title + "," + task.dueDate + "," +
task.isCompleted);
            }
        } catch (IOException e) {
            System.out.println("Error saving tasks: " + e.getMessage());
        }
    }

    private static void loadTasksFromFile() {
        File file = new File(FILE_NAME);
        if (!file.exists()) return;
```

```
        try (Scanner fileScanner = new Scanner(file)) {
            while (fileScanner.hasNextLine()) {
                String[] data = fileScanner.nextLine().split(",");
                if (data.length == 3) {
                    Task task = new Task(data[0], data[1]);
                    task.isCompleted = Boolean.parseBoolean(data[2]);
                    tasks.add(task);
                }
            }
        } catch (IOException e) {
            System.out.println("Error loading tasks: " + e.getMessage());
        }
    }
}
```