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

/**
*  Simple To-Do List Application
*  ----------------------------  * Features:
*  - Add new tasks
*  - View all tasks
*  - Mark tasks as complete
*  - Delete tasks
*  - Save/Load tasks from a text file
 */
public class ToDoListApp {

   private static final String FILE_NAME = "tasks.txt";
private static Scanner sc = new Scanner(System.in);
private static ArrayList<Task> tasks = new ArrayList<>();

   public static void main(String[] args) {
loadTasks();       int choice;       do {
         System.out.println("\n====== TO-DO LIST MENU ======");
         System.out.println("1. Add Task");
         System.out.println("2. View Tasks");
         System.out.println("3. Mark Task as Done");
         System.out.println("4. Delete Task");
         System.out.println("5. Save Tasks");
```

```
        System.out.println("6. Exit");
        System.out.print("Enter your choice: ");
choice = getInt();

        switch (choice) {
case 1: addTask(); break;
case 2: viewTasks(); break;
case 3: markDone(); break;
case 4: deleteTask(); break;
case 5: saveTasks(); break;
case 6:
                saveBeforeExit();
                System.out.println("Exiting program. Goodbye!");
break;
            default: System.out.println("Invalid choice. Try again.");
        }
    } while (choice != 6);
  }

  // --------- Functions ----------
private static void addTask() {
    System.out.print("Enter task title: ");
    String title = sc.nextLine().trim();
    System.out.print("Enter description (optional): ");
    String desc = sc.nextLine().trim();
```

```
      Task t = new Task(title, desc);
tasks.add(t);
      System.out.println("Task added successfully!");
   }

   private static void viewTasks() {
if (tasks.isEmpty()) {
         System.out.println("No tasks yet!");
         return;
      }
      System.out.println("\n--- TASK LIST ---");
for (int i = 0; i < tasks.size(); i++) {
         Task t = tasks.get(i);
         String status = t.isDone ? "[✓]" : "[ ]";
         System.out.println((i + 1) + ". " + status + " " + t.title);
if (!t.description.isEmpty())
            System.out.println("   → " + t.description);
      }
   }

   private static void markDone() {
viewTasks();       if
(tasks.isEmpty()) return;
System.out.print("Enter task
number to mark as done: ");
int num = getInt();       if (num < 1
```

```
|| num > tasks.size()) {
System.out.println("Invalid task
number!");          return;
      }
      tasks.get(num - 1).isDone = true;
      System.out.println("Task marked as done!");
   }

   private static void deleteTask() {
viewTasks();       if
(tasks.isEmpty()) return;

      System.out.print("Enter task number to delete: ");
int num = getInt();       if (num < 1 || num > tasks.size())
{          System.out.println("Invalid task number!");
return;
      }
      tasks.remove(num - 1);
      System.out.println("Task deleted.");
   }

   private static void saveTasks() {
      try (PrintWriter pw = new PrintWriter(new FileWriter(FILE_NAME))) {
for (Task t : tasks)
            pw.println(t.title + "|" + t.description + "|" + t.isDone);
         System.out.println("Tasks saved to " + FILE_NAME);
```

```
    } catch (IOException e) {
      System.out.println("Error saving: " + e.getMessage());
    }
  }

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

    try (Scanner f = new Scanner(file)) {
while (f.hasNextLine()) {
        String[] p = f.nextLine().split("\\|");
if (p.length >= 3)
          tasks.add(new Task(p[0], p[1], Boolean.parseBoolean(p[2])));
      }
      System.out.println("Loaded " + tasks.size() + " tasks.");
    } catch (IOException e) {
      System.out.println("Error loading file: " + e.getMessage());
    }
  }

  private static void saveBeforeExit() {
    System.out.print("Save tasks before exit? (y/n): ");
String ans = sc.nextLine().toLowerCase();      if
(ans.equals("y")) saveTasks();
  }
```

```
    private static int getInt() {
        try {
            return Integer.parseInt(sc.nextLine());
        } catch (Exception e) {
return -1;
        }
    }

    // --------- Task Class ----------
static class Task {
        String title;
        String description;
boolean isDone;

        Task(String title, String desc) {
this.title = title;
this.description = desc;
this.isDone = false;
        }

        Task(String title, String desc, boolean done) {            this.title = title;
this.description = desc;            this.isDone = done;
        }
    }
}
```