

**Final Project (Groups of 1-2 people allowed. Please see the requirements for each team size in section 4.0)**

## **1.0 Introduction**

*HummingBee Home Garden Sprinkler System* is an automated, microprocessor-based watering system for lawns and gardens. The system can be programmed to water the lawns and gardens at specified times and/or based on the daily temperatures.

Using the OO Analysis and Design techniques, develop an Object Model of the *HummingBee Garden Sprinkler System* and implement it in Java. The implementation is a computer-based simulation of the Sprinkler System, where the various sensors, their functionality, the control panel to arm and program the system are simulated using a Graphic User Interface.

## **2.0 Functionality**

The *Home Garden Sprinkler System* consists of a software module which controls and monitors a number of sprinklers and temperature sensors, installed in a yard. The home-owner (client) interacts with the system through a control panel with a keypad and a small display screen. The home owner uses the control panel to program and configure the system to schedule the sprinklers in a specific area to start at a specific time(s) on specific days(s) of the week. The system can be enabled, disabled and configured for a week at a time. If the configuration is not changed, the weekly pattern is followed regularly. The system can also be configured to start when the day temperatures reach a certain limit.

The system is connected to a number of sprinklers located 10 ft apart and are grouped by their location as North, South, East and West groups, where each sprinkler has an Id number, unique to its group. For example, the sprinkler 3N belongs to the North group.

The system can be configured to start each individual sprinkler or a group of sprinklers at specific times of the day. Each sprinkler or a group of sprinklers can be configured to regulate the volume of water flow per hour.

All interaction with *HummingBee Sprinkler system* is managed by a user-interaction subsystem that reads the input provided through the keypad with function keys. The system can be configured by setting a pattern of usage which consists of the id of the sprinkler, the start time and stop time on each day. Additionally, the system can be configured to override the daily schedule if the temperatures fall below certain limit or rise above a certain limit. For example, the sprinklers should not start if the temperature is below 55F and the duration and the frequency of watering can be increased if the temperature rises above 90 F.

The home-owner can select to view the status of the sprinklers, where the status for each sprinkler is shown on the screen by its group name, id and status OK (functional) or NOTOK (not functional), ON (currently on) and NOTON (currently not on). The home-owner can, at any time, enable or disable a

specific sprinkler for a period of time. The system enables the home-owner to check the total water consumption by the sprinklers per month, which is shown on the display screen, using a visualization tool like a bar graph.

**The functionality of the system that you should implement is described below:**

- 2.1 Program the system (with a weekly schedule) to start the sprinklers at designated times.
- 2.2 Enable and disable the system.
- 2.3 Show the activation of the sprinklers.
- 2.4 Display the status of the sprinklers.
- 2.5 Display the total water usage.
- 2.6 Program the system to activate/deactivate the sprinklers based on temperatures.
- 2.7 Adjust the temperatures to show the activation/deactivation of the sprinklers.
- 2.8 Program the system to activate/deactivate the sprinkler groups (and individual sprinklers).
- 2.9 Display a map/schematic of the garden layout and the location of sprinklers. Display a graph showing the water usage by the sprinkler groups.

### **3.0 Deliverables**

This project is deliverable in two parts:

**Part 1 (Analysis and Design):** The Analysis and Design documents are to be submitted and should include the following:

- **Use-Cases** to illustrate the functionality of the system. Include any screen shots that you may find useful to illustrate the functionality.
- **CRC cards** to show a list of the important classes and a brief description of their responsibilities.
- Interaction Diagrams (Use **Sequence diagrams**) for the most important Use cases and State Transition Diagrams (where applicable).

**Part 2 (Implementation):**

- A detailed **class diagram**
- A **package diagram**
- Source code
- **Demo to class** (10th week of classes)

## **4.0 Team Requirements**

4.1 Requirements for 1-person teams: Sections 2.1 to 2.8 should be implemented.

4.2 Requirements for 2-person teams: Sections 2.1 to 2.9 should be implemented.