Class Outline

Jared Nixon

**Table of Contents**

1. Table of Contents . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
2. System Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
3. Class Simulation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
4. Class SensorMount. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
5. Class Sensor . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
6. Class SensorFactory. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .6
7. Class AirFactory . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
8. Class EarthFactory . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
9. Class WaterFactory. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
10. Class AirSensor . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
11. Class EarthSensor. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
12. Class WaterSensor . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
13. Class Display . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
14. Class DisplayFactory . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
15. Class CRT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
16. Class Plotter . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
17. Class Meter . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
18. System Overview

1.1.1 The functionality of this system will simulate sensors reading information from the environment and relaying the readings back to the user. The user will also be allowed to subscribe and unsubscribe sensors via pressing the ‘s’ key. This simulation will be generated from an XML file, which will contain all information about the display devices and sensors, including how many of each and their respective types. The class outline below will help illustrate the functionality of this system and list some of the intended responsibilities of each designed object.

2.0 Object Description

2.1 Class **Simulation**

2.1.1 Member Variables

crtDisplay – Protected – Pointer to a CRT object.

plotterDisplay – Protected – Pointer to a Plotter object.

meterDisplay – Protected – Pointer to a Meter object.

tempSensor – Private – Pointer to a temporary sensor object used to store data until type of sensor is decided.

tempDisplay – Private – Pointer to a temporary display object used to store data until type of display is decided.

airSensor – Private – Pointer to a Air Sensor object.

earthSensor – Private – Pointer to a Earth Sensor object.

waterSensor – Private – Pointer to a Water Sensor object.

sensorMount – Private – Pointer to the Sensor Mount object.

sensorContainer – Private – Vector container used to store all sensors created by Simulation.

displayContainer – Private – Vector container used to store all display devices created by Simulation.

Hit – Private – key hit flag.

Ch – Private – Character key which was hit.

Disp – Display device holding the sensor to be toggled.

Sens – Sensor to unsubscribe or resubscribe.

2.1.2 Member Functions

2.1.2.1 Function Simulation - SINGLETON

Actions Performed – Creates the sensorRack object, as well as the temporary Display and Sensor.

Arguments – N/A

Return Value – N/A

2.1.2.2 Function ~Simulation

Actions Performed – Deletes all objects created when the user ends the simulation.

Arguments – N/A

Return Value – N/A

2.1.2.3 Function initializeSimulation

Actions Performed – Creates data parser and passes the file name of the XML file to be read. Creates all and Display objects specified by the XML file. Stores these objects in appropriate vector containers.

Arguments – N/A

Return Value – Void

2.1.2.4 Function runSimulation

Actions Performed – Creates the timer loop so that data may be sent to the output on five second intervals. Watches keyboard input to see when the user wants to end the simulation. Also watches input to see if the key ‘s’ is pressed, and when it is pressed, pulls up the menu so that the user may monitor / unmonitor a specific sensor in a display device.

Arguments – N/A

Return Value – Void

2.2.2.5 Function getInstance

Actions Performed – Gets the instance of a simulation object, if not one, will create one and only one.

Arguments – N/A

Return Value – Simulation\* (Pointer to a simulation object)

2.2 Class **SensorMount**

2.2.1 Member Variables

Data – Private – Holds the randomly generated integer to relay to the display object.

2.2.2 Member Functions

2.2.2.1 Function SensorMount - SINGLETON

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.2.2.2 Function ~SensorMount

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.2.2.3 Function propogateData

Actions Performed – Matches the sensor ID to a display ID, when matched, will call the update function to let the observer know that it needs to be updated. Will also only call update function if the sensor is being monitored. (based on the bool true or false)

Arguments – displayContainer (vector), sensorContainer (vector)

Return Value – Void

2.2.2.5 Function getInstance

Actions Performed – Gets the instance of a Sensor Mount object, if not one, will create one and only one.

Arguments – N/A

Return Value – SensorMount\* (Pointer to a Sensor Mount object)

2.3 Class **Sensor**

2.3.1 Member Variables

m\_chType[32] – Protected – Holds the type of sensor.

m\_chMaterial[32] – Protected – Holds the material being detected by the sensor.

m\_iSensorID – Protected – Holds the ID value of the sensor.

m\_chUnits[32] – Protected – Holds the units that the sensors data will be in.

m\_dValueMax – Protected – Holds the max value that can be generated (specific to each type of sensor).

m\_dValueMin – Protected – Holds the min value that can be generated (specific to each type of sensor).

2.3.2 Member Functions

2.3.2.1 Function Sensor

Actions Performed – Will only seed the random number generated so that it may be used.

Arguments – N/A

Return Value – N/A

2.3.2.2 Function ~Sensor

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.3.2.3 Function setType

Actions Performed – Copies the type into the private variable m\_chType of the object.

Arguments – char\* (array)

Return Value – Void

2.3.2.4 Function setMaterial

Actions Performed – Copies the material the sensor is indicating into the private variable m\_chMaterial of the object.

Arguments – char\* (array)

Return Value – Void

2.3.2.5 Function setSensorID

Actions Performed – Copies the sensor ID into the private variable m\_iSensorID of the object.

Arguments – int\* (array)

Return Value – Void

2.3.2.6 Function setUnits

Actions Performed – Copies the units the data is in into the private variable m\_chUnits of the object.

Arguments – char\* (array)

Return Value – Void

2.3.2.7 Function setValueMax

Actions Performed – Copies the max value possible to be generated into the private variable m\_dValueMax of the object.

Arguments – double

Return Value – Void

2.3.2.8 Function setValueMin

Actions Performed – Copies the min value possible to be generated into the private variable m\_dValueMin of the object.

Arguments – double

Return Value – Void

2.3.2.9 Function getType

Actions Performed – Gets the type of sensor from the data parser.

Arguments – N/A

Return Value – char\* (array)

2.3.2.10 Function getMaterial

Actions Performed – Gets the type of material the sensor is indicating from the data parser.

Arguments – N/A

Return Value – char\* (array)

2.3.2.11 Function getSensorID

Actions Performed – Gets the sensor ID from the data parser.

Arguments – N/A

Return Value – int\* (array)

2.3.2.12 Function getUnits

Actions Performed – Gets the units the data is in from the data parser.

Arguments – N/A

Return Value – char\* (array)

2.3.2.13 Function getValueMax

Actions Performed – Gets the max value possible to be generated (sensor specific).

Arguments – N/A

Return Value – double\* (pointer)

2.3.2.14 Function getValueMin

Actions Performed – Gets the min value possible to be generated (sensor specific).

Arguments – N/A

Return Value- double\* (pointer)

2.3.2.15 Function GenerateData

Actions Performed – Generates the integer data associated with the sensor. These values with range from m\_dValueMax to m\_dValueMin.

Arguments – max (double), min (double)

Return Value – int

2.4 class **SensorFactory - <<interface>>**

2.15.1 Member Variables

*None, simply an interface for all of the factory sub classes.*

2.15.2 Member Functions

*None, simply an interface for all of the factory sub classes.*

2.5 class **AirFactory**

2.15.1 Member Variables

*None, just used to create the Air Sensor objects when needed.*

2.15.2 Member Functions

2.14.2.1 AirFactory - SINGLETON

Actions Performed – Private constructor for the AirFactory object.

Arguments – Display\*(Pointer to a factory object)

Return Value – N/A

2.14.2.2 ~AirFactory

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.14.2.3 CreateSensor

Actions Performed – Creates the airSensor object and returns it.

Arguments – Sensor\*(Pointer to a Sensor object)

Return Value – AirSensor\*(Pointer to an airSensor object)

2.14.2.4 getInstance

Actions Performed – Gets the instance of a AirSensor factory object, if one isnt made, one is created.

Arguments – N/A

Return Value – AirFactory\* (Pointer to a Air factory object)

2.6 class **EarthFactory**

2.15.1 Member Variables

2.15.2 Member Functions

2.14.2.1 AirFactory - SINGLETON

Actions Performed – Private constructor for the EarthFactory object.

Arguments – N/A

Return Value – N/A

2.14.2.2 ~AirFactory

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.14.2.3 CreateSensor

Actions Performed – Creates the earthSensor object and returns it.

Arguments – Sensor\*(Pointer to a Sensor object)

Return Value – EarthSensor\*(Pointer to an earthSensor object)

2.14.2.4 getInstance

Actions Performed – Gets the instance of a EarthSensor factory object, if one isnt made, one is created.

Arguments – N/A

Return Value – EarthFactory\* (Pointer to a Earth Factory object)

2.7 class **WaterFactory**

2.15.1 Member Variables

2.15.2 Member Functions

2.14.2.1 AirFactory - SINGLETON

Actions Performed – Private constructor for the WaterFactory object.

Arguments – N/A

Return Value – N/A

2.14.2.2 ~AirFactory

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.14.2.3 CreateSensor

Actions Performed – Creates the waterSensor object and returns it.

Arguments – Sensor\*(Pointer to a Sensor object)

Return Value – WaterSensor\*(Pointer to an WaterSensor object)

2.14.2.4 getInstance

Actions Performed – Gets the instance of a WaterSensor factory object, if one isnt made, one is created.

Arguments – N/A

Return Value – WaterFactory\* (Pointer to a WaterFactory object)

2.8 Class **AirSensor**

2.4.1 Member Variables

*All variables used are inherited from parent class Sensor (see above).*

2.4.2 Member Functions

2.4.2.1 Function AirSensor

Actions Performed – Associates all data from a tempSensor with the data for the airSensor. This includes setting all six variables (see private variables of Sensor).

Arguments – Sensor\* (Pointer to the tempSensor from Simulation)

Return Value – N/A

2.4.2.2 Function ~AirSensor

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

Return Value – N/A

2.9 Class **EarthSensor**

2.4.1 Member Variables

*All variables used are inherited from parent class Sensor (see above).*

2.4.2 Member Functions

2.4.2.1 Function EarthSensor

Actions Performed – Associates all data from a tempSensor with the data for the earthSensor. This includes setting all six variables (see private variables of Sensor).

Arguments – Sensor\* (Pointer to the tempSensor from Simulation)

Return Value – N/A

2.4.2.2 Function ~EarthSensor

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.10 Class **WaterSensor**

2.4.1 Member Variables

*All variables used are inherited from parent class Sensor (see above).*

2.4.2 Member Functions

2.4.2.1 Function WaterSensor

Actions Performed – Associates all data from a tempSensor with the data for the waterSensor. This includes setting all six variables (see private variables of Sensor).

Arguments – Sensor\* (Pointer to the tempSensor from Simulation)

Return Value – N/A

2.4.2.2 Function ~WaterSensor

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.11 Class **Display**

2.10.1 Structures

Toggle – Structure used to associated a BOOL with each Sensor (to determine is monitor or not monitored).

2.10.1.1 Structure Variables

2.10.2 Member Variables

m\_chType[32] – Private – Holds the type of display device.

m\_iID – Private – Holds the ID value of the display device.

m\_iCountID – Private – Holds the number of sensors the display device recieves data from.

2.10.3 Member Functions

2.10.2.1 Function Display

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.10.2.2 Function ~Display

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.10.2.3 Function Update

Actions Performed – None. Strictly a virtual function so children classes can implement their own version of the Update function.

Arguments – Void

Return Value – Void

2.10.2.4 Function setDisplayType

Actions Performed – Copies the display type into the private variable m\_chType of the object.

Arguments – char\* (array)

Return Value – Void

2.10.2.5 Function setDisplayID

Actions Performed – Copies the display ID into the private variable m\_iID of the object.

Arguments – int\* (array)

Return Value – Void

2.10.2.6 Function setDisplayCountID

Actions Performed – Copies the display count into the private variable m\_iCountID of the object.

Arguments – int\* (Pointer to an integer)

Return Value – Void

2.10.2.7 Function getDisplayType

Actions Performed – Gets the display type from the data parser.

Arguments – N/A

Return Value – char\* (array)

2.10.2.8 Function getDisplayID

Actions Performed – Gets the display ID from the data parser.

Arguments – N/A

Return Value – int\* (array)

2.10.2.9 Function getDisplayCountID

Actions Performed – Gets the display count from the data parser.

Arguments – N/A

Return Value – int\* (Pointer to an integer)

2.10.2.10 Function InitStruct

Actions Performed – Initializes all bools for each displayID.

Arguments – void

Return Value – void

2.10.2.11 Function getStructure

Actions Performed – Gets the structure associated with each Display Device.

Arguments – N/A

Return Value – Structure

2.12 Class **DisplayFactory**

2.14.1 Member Variables

crtDisplay – Private – Pointer to a CRT Display object.

meterDisplay – Private – Pointer to a Meter Display object.

plotterDisplay – Private – Pointer to a Plotter Display object.

2.14.2 Member Functions

2.14.2.1 DisplayFactory - SINGLETON

Actions Performed – Private constructor for the DisplayFactory object.

Arguments – Display\*(Pointer to a factory object)

Return Value – N/A

2.14.2.2 ~DisplayFactory

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.14.2.3 CreateDisplay

Actions Performed – Creates the display object according to its type and returns it.

Arguments – Display\*(Pointer to a display object)

Return Value – Display\*(Pointer to a display object)

2.14.2.4 getInstance

Actions Performed – Gets the instance of a display factory object, if one isnt made, one is created.

Arguments – N/A

Return Value – DisplayFactory\* (Pointer to a display factory object)

2.13 Class **CRT**

2.11.1 Member Variables

*All variables used are inherited from parent class Display (see above).*

2.11.2 Member Functions

2.11.2.1 Function CRT

Actions Performed - Associates all data from a tempDisplay with the data for the crtDisplay. This includes setting all three variables (see private variables of Display).

Arguments - Display\* (Pointer to the tempDisplay from Simulation)

Return Value – N/A

2.11.2.2 Function ~CRT

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.11.2.3 Function Update

Actions Performed – Outputs information to the user according the what type of sensor is passed to it. The observer will also generate data when it is time to be updated. (instead of data being pushed to it)

Arguments – Sensor\* (Pointer to a sensor object)

Return Value – void

2.14 Class **Plotter**

2.12.1 Member Variables

*All variables used are inherited from parent class Display (see above).*

2.12.2 Member Functions

2.12.2.1 Function Plotter

Actions Performed – Associates all data from a tempDisplay with the data for the plotterDisplay. This includes setting all three variables (see private variables of Display).

Arguments – Display\* (Pointer to the tempDisplay from Simulation)

Return Value – N/A

2.12.2.2 Function ~Plotter

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.12.2.3 Function Update

Actions Performed – Outputs information to the user according the what type of sensor is passed to it. The observer will also generate data when it is time to be updated. (instead of data being pushed to it)

Arguments – Sensor (Pointer to a sensor object)

Return Value – void

2.15 Class **Meter**

2.13.1 Member Variables

*All variables used are inherited from parent class Display (see above).*

2.13.2 Member Functions

2.13.2.1 Function Meter

Actions Performed – Associates all data from a tempDisplay with the data for the meterDisplay. This includes setting all three variables (see private variables of Display).

Arguments – Display\* (Pointer to the tempDisplay from Simulation)

Return Value – N/A

2.13.2.2 Function ~Meter

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.13.2.3 Function Update

Actions Performed – Outputs information to the user according the what type of sensor is passed to it. The observer will also generate data when it is time to be updated. (instead of data being pushed to it)

Arguments – Sensor(Pointer to a sensor object)

Return Value – void

Notes\*\*

* Inheritance is used for the vast majority of variables related to specific sensors and displays.
* The distinct classes for each type of sensor and display may not seem to do much, but making them loosely coupled is necessary for possible future designs, therefore all variables are just being inherited because of the uniformity among the sensors and displays for the current assignment.