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 Cadmium . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
7. Class CarbonMonoxide . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
8. Class Chlorine . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
9. Class NuclearWaste. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
10. Class OilResidue . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
11. Class RawSewage . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
12. Class Display . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
13. Class CRT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
14. Class Plotter . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
15. Class Meter . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
16. System Overview

1.1.1The functionality of this system will simulate sensors reading information from the environment and relaying the readings back to the user. 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.

cadmiumSensor – Private – Pointer to a Cadmium Sensor object.

nuclearSensor – Private – Pointer to a Nuclear Sensor object.

monoxideSensor – Private – Pointer to a Monoxide Sensor object.

chlorineSensor – Private – Pointer to a Chlorine Sensor object.

residueSensor – Private – Pointer to a Residue Sensor object.

sewageSensor – Private – Pointer to a Sewage Sensor object.

sensorRack – 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.

2.1.2 Member Functions

2.1.2.1 Function Simulation

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.

Arguments – N/A

Return Value – Void

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

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 recieveData

Actions Performed – Calls function used to generate a random integer to be used as the data.

Arguments – displayContainer (vector), sensorContainer (vector)

Return Value – Void

2.2.2.4 Function sendData

Actions Performed – Relays data generated by recieveData to the Display object to be displayed.

Arguments – displayContainer (vector), Sensor (One at a time), data

Return Value – Void

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 **Cadmium**

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 Cadmium

Actions Performed – Associates all data from a tempSensor with the data for the cadmiumSensor. 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 ~Cadmium

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.5 Class **CarbonMonoxide**

2.5.1 Member Variables

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

2.5.2 Member Functions

2.5.2.1 Function CarbonMonoxide

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

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

Return Value – N/A

2.5.2.2 Function ~CarbonMonoxide

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.6 Class **Chlorine**

2.6.1 Member Variables

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

2.6.2 Member Functions

2.6.2.1 Function Chlorine

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

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

Return Value – N/A

2.6.2.2 Function ~Chlorine

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.7 Class **NuclearWaste**

2.7.1 Member Variables

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

2.7.2 Member Functions

2.7.2.1 Function NuclearWaste

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

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

Return Value – N/A

2.7.2.2 Function ~NuclearWaste

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.8 Class **OilResidue**

2.8.1 Member Variables

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

2.8.2 Member Functions

2.8.2.1 Function OilResidue

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

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

Return Value – N/A

2.8.2.2 Function ~OilResidue

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.9 Class **RawSewage**

2.9.1 Member Variables

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

2.9.2 Member Functions

2.9.2.1 Function RawSewage

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

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

Return Value – N/A

2.9.2.2 Function ~RawSewage

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

2.10 Class **Display**

2.10.1 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.

2.10.2 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 – Sends data recieved from Sensor Mount to the display terminal for the user to see. The function will take one display device and one sensor that have already been matched and are ready for output, along with the data to be sent to the output.

Arguments – Display\* (Pointer to a display device), Sensor\* (Pointer to a sensor), int data

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)

2.11 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

2.12 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

2.13 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 Display

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 ~Display

Actions Performed – N/A

Arguments – N/A

Return Value – N/A

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.