Implementation (Pseudo Code)

Class: DataType

```
enum DataType
    0 = ts
    1 = device
    2 = co
    3 = humidity
    4 = light
    5 = lpg
    6 = motion
    7 = smoke
    8 = temp
```

Class: DataRecord

```
PRIVATE:
    DOUBLE timestamp
    STRING deviceID
    DOUBLE co
    DOUBLE humidity
    BOOL light
    DOUBLE lpg
    BOOL motion
    DOUBLE smoke
    DOUBLE temperature
PUBLIC:
    CONSTRUCTOR
        SET all private members

    DOUBLE FUNC “getTs()”
        RETURN timestamp

    STRING FUNC “getDevice()”
        RETURN deviceID

    DOUBLE FUNC “getCo()”
        RETURN co
```

```
DOUBLE FUNC “getHumidity()”
    RETURN humidity

BOOL FUNC “getLight()”
    RETURN light

DOUBLE FUNC “getLpg()”
    RETURN lpg

BOOL FUNC “getMotion()”
    RETURN motion

DOUBLE FUNC “getSmoke()”
    RETURN smoke

DOUBLE FUNC “getTemp()”
    RETURN temperature
```

Class: DataRecord

```
VOID FUNC “setTs(double ts)”
    this.ts = ts

VOID FUNC “setDevice(string device)”
    this.device = device

VOID FUNC “setCo(double co)”
    this.co = co

VOID FUNC “setHumidity(double humidity)”
    this.humidity = humidity

VOID FUNC “setLight(bool light)”
    this.light=light

VOID FUNC “setLpg(double lpg)”
    this.lpg = lpg

VOID FUNC “setMotion(bool motion)”
    this.motion=motion
```

```
VOID FUNC “setSmoke(double smoke)”
    this.smoke=smoke

VOID FUNC “setTemp(double temp)”
    Rthis.temp = temp
```

Class: DataRead

```
PRIVATE:
     STRING fileName
     LIST DataRecord dataPoints

PUBLIC:
     NONE FUNC “readData(STRING filename)”
          SET fileName to filename
          IF Open file fails:
               RETURN “ERROR”
          ELSE:
               For each line, get each value
               Create DataRecord instance with values
               Append instance to dataPoints list
               Close file
               RETURN nothing

     dataRecord FUNC “getDataRecord(INT index)”
          RETURN dataPoints(INT index)

     LIST DataRecord FUNC “getDataRecords()”
          RETURN dataPoints
```

Class: DataStorage

```
PRIVATE:
    LIST DataRead dataFileReadings

PUBLIC:
    NONE FUNC “storeData(DataRead DataReading)”
        Append DataReading to dataFileReadings List
        RETURN nothing

    DataRead FUNC “extractData(INT index)”
        RETURN dataFileReadings(INT index)
```

Class: Hub

```
PRIVATE:
    DataRead dataPoints

PUBLIC:
    NONE FUNC “processData(DataRead datapoints)”
        SET dataPoints to datapoints
        INT i
        SET i to 0
        WHILE i < LENGTH of dataPoints.getDataRecords:
            dataPoints.getDataRecord(i).SetTemp((item.getDataRecord(i)
            .getTemp()-32)*(5/9))
            INCREMENT i by 1
        RETURN nothing


    DataRead FUNC “exportData()”
        RETURN dataPoints
```

Class: DataWrite

```
PRIVATE:
    STRING fileName

PUBLIC:
    NONE FUNC “writeData(STRING filename,DataRead dataread,INT datatype)”
        SET fileName to filename
        Create file named fileName
        IF Open file fails:
            RETURN “ERROR”
        ELSE:
            INT i
            SET i to 0
            WHILE i < LENGTH of dataread.getDataRecords:
                WRITE dataread.getDataRecord(i).getTs()
                WRITE dataread.getDataRecord(i).getDevice()
                WRITE dataread.getDataRecord(i).get”datatype”()
                //Case statement required to choose datatype

            Close file
            RETURN nothing
```

Main

```
DataStorage MainStorage
Hub MainHub
DataRead Read1

Read1.readData(“iot_telemetry_data.csv”)

MainStorage.storeData(Read1)

MainHub.processData(MainStorage.extractData(0))

INT j
SET j to 2
WHILE j < 8:
    writeData(“fileExport”+j-1, MainHub.exportData(),  dataType)
    INCREMENT j by 1
```