Practical 3 (due 2023-08-11 @ 09:00)

The purpose of this practical is for you to become familiar with the basics of composition and delegation.

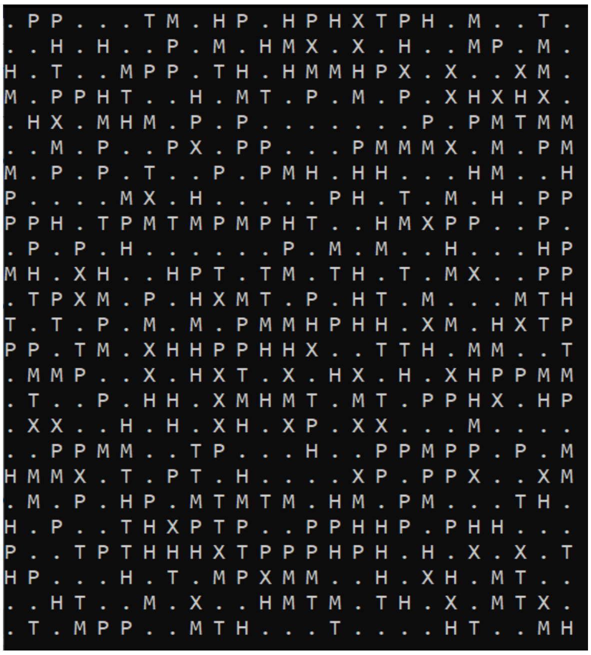
You may use the memo for Practical 2 in the creation of Practical 3. This will not be considered as plagiarism if you include a comment indicating you have done so.

Modify the code from Practical 2 as follows:

- Remove the Sensor ID from the `IIoTSensor` record structure.
- Add a new variable of type bool to `IIoTSensor` representing whether the sensor is active. When an `IIoTMonitor` object is created, every sensor must be set to active.
- The range of the Light Intensity must now be [10000, 30000].

Create a class called `IIoTScanner`

- This class includes a public void member function called `scan()`
    - The member function must take an `IIoTMonitor` reference as input
    - The member function must loop through all the sensors in the array and check if any of the sensors have a reading of 0 for either the temperature, humidity, pressure or light intensity. If any of the readings is 0 for a particular sensor, the sensor must be set as inactive.
- The `IIoTMonitor` must be updated so that
    - it **has-a**(n) `IIoTScanner` whose lifecycle it manages directly (the `IIoTScanner` must be instantiated from the free-store when the `IIoTMonitor` is created and de-allocated when the `IIoTMonitor` is destroyed).
    - it has a `scan()` member function which it delegates to its contained `IIoTScanner`
    - Note that the `IIoTMonitor` needs to know about the `IIoTScanner` and the reverse is true. This can cause a problem with compilation that needs to be solved using forward declarations. Please see the following video that addresses such a problem: https://youtu.be/WTQP0JQ7tBY.
    - The `toString()` member function must be updated so that inactive sensors are represented by the character '**X**'.
- The following screenshot shows an example output after scanning the IIoT environment:



```
. P P . . . T M . H P . H P H X T P H . M . . T .
. . H . H . . P . M . H M X . X . H . . M P . M .
H . T . . M P P . T H . H M M H P X . X . . X M .
M . P P H T . . H . M T . P . M . P . X H X H X .
. H X . M H M . P . P . . . . . . . P . P M T M M
. . M . P . . P X . P P . . . P M M M X . M . P M
M . P . P . T . . P . P M H . H H . . . H M . . H
P . . . . M X . H . . . . . P H . T . M . H . P P
P P H . T P M T M P M P H T . . H M X P P . . P .
. P . P . H . . . . . . P . M . M . . H . . . H P
M H . X H . . H P T . T M . T H . T . M X . . P P
. T P X M . P . H X M T . P . H T . M . . . M T H
T . T . P . M . M . P M M H P H H . X M . H X T P
P P . T M . X H H P P H H X . . T T H . M M . . T
. M M P . . X . H X T . X . H X . H . X H P P M M
. T . . P . H H . X M H M T . M T . P P H X . H P
. X X . . H . H . X H . X P . X X . . . M . . . .
. . P P M M . . T P . . . H . . P P M P P . P . M
H M M X . T . P T . H . . . . X P . P P X . . X M
. M . P . H P . M T M T M . H M . P M . . . T H .
H . P . . T H X P T P . . P P H H P . P H H . . .
P . . T P T H H H X T P P P H P H . H . X . X . T
H P . . . H . T . M P X M M . . H . X H . M T . .
. . H T . . M . X . . H M T M . T H . X . M T X .
. T . M P P . . M T H . . . T . . . . H T . . M H
```

Upload and submission

- Create a PDF design document named **Design.pdf**
- When your program is working and you have created a design document, you must add your work to an archive file in the **zip** compression format. The name of the archive must be in the following format:

  `SURNAME_INITIALS_STUDENTNUMBER_CSC01B1_2023_P0.zip`

  e.g. for a student called Anne Student with student number 123456789

  `STUDENT_A_123456789_CSC01B1_2023_P0.zip`
- The archive must contain the following directories / folders:
  - **src** - containing the C++ source code needed to compile your program
  - **bin** – containing an executable generated from your source code
  - **doc** – containing your design document

	Mark sheet	
	Design	10
	IIoTScanner class with scan member function	10
	Containment relationship	10
	Allocation of contained class	10
	De-allocation of contained class	10
	Delegation of scan	10
	Demonstration of functionality in a main function	10
	Total	/70