COMP1406 - Assignment #1

(Due: **Monday, January 23rd @ 12 noon**)

In this assignment, you will create some simple programs in JAVA to get used to the language constructs that you learned last term, including IF statements, FOR loops and arrays.

(1) Water Level Program

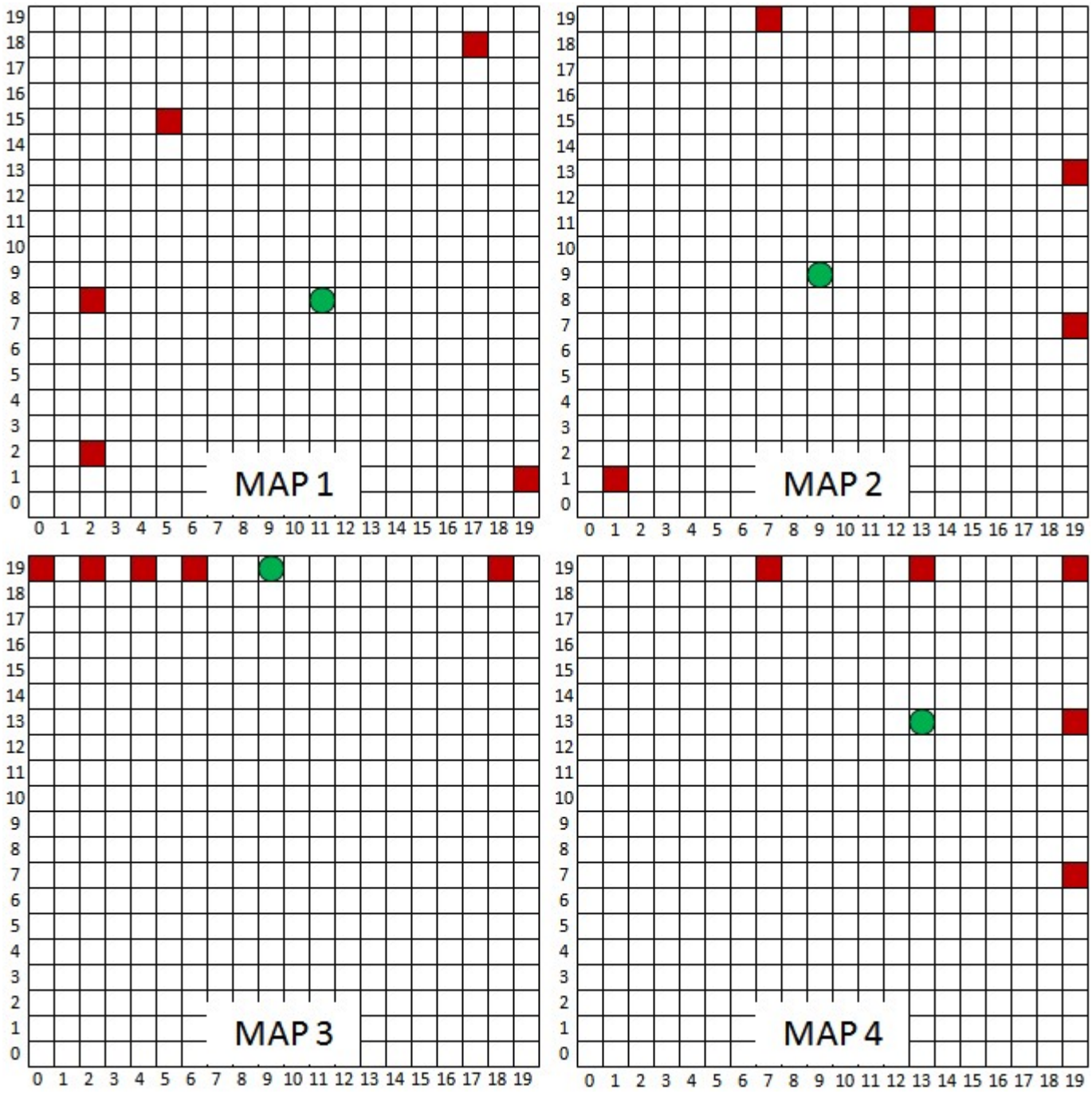
Write a class called **WaterLevelProgram** that simulates a flood sensor in a basement of a house. The sensor is to watch the water level rise continuously and then it should detect when the flood has subsided by detecting a decrease in the water level. To simulate this, the user will type in the level of water in mm. The program should simply keep asking for the latest water level (as a sensor would sample constantly). Once the program has detected three decreasing readings in a row, it should display a message indicating that the flood has subsided and then quit. Here is example output:

```
What is the water level at now (in mm): 20
What is the water level at now (in mm): 21
What is the water level at now (in mm): 23
What is the water level at now (in mm): 25
What is the water level at now (in mm): 27
What is the water level at now (in mm): 27
What is the water level at now (in mm): 34
What is the water level at now (in mm): 22
What is the water level at now (in mm): 25
What is the water level at now (in mm): 26
What is the water level at now (in mm): 28
What is the water level at now (in mm): 27
What is the water level at now (in mm): 26
What is the water level at now (in mm): 25

It appears that the flood is subsiding.
```

(2) Hospital Builder Program

On the next page, there are 4 grid-based maps each with 5 small towns shown as red squares. We would like to write a program that determines the "optimal" grid location to build a hospital so that it lies at a distance that minimizes the travel from each town. That is ... the furthest distance that a town resident has to travel in order to reach the hospital is kept to a minimum. The answer is shown as a green circle for each of the 4 maps.



Write a class called **HospitalBuilderProgram** that computes the green circle location for each map. The program must read in a 3-dimensional array of town locations as shown below:

```
private static byte[][][]   maps = {{{2, 2}, {2, 8}, {5, 15}, {19, 1}, {17, 17}},
                                    {{1, 1}, {7, 19}, {13, 19}, {19, 7}, {19, 13}},
                                    {{0, 19}, {2, 19}, {4, 19}, {6, 19}, {18, 19}},
                                    {{7, 19}, {13, 19}, {19, 19}, {19, 13}, {19, 7}}};
```

The array has 4 maps. Each map has 5 towns. Each town has an (x,y) coordinate that matches the 4 maps shown earlier. Your program must loop through these towns and compute the answer to the problem. The easiest way is to compute the distance to each town from every one of the 20x20 grid locations and keep the best one as the answer. The distance between points (x_1, y_1) and (x_2, y_2) can be computed as **d** = $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. The program should print out the (x, y) answer for each of the 4 maps.

IMPORTANT SUBMISSION INSTRUCTIONS:

Submit your ZIPPED **IntelliJ project file** as you did during the first tutorial for assignment 0.

- YOU WILL LOSE MARKS IF YOU ATTEMPT TO USE ANY OTHER COMPRESSION FORMATS SUCH AS **.RAR**, **.ARC, .TGZ**, **.JAR**, **.PKG**, **.PZIP**.

- If your internet connection at home is down or does not work, we will not accept this as a reason for handing in an assignment late ... so make sure to submit the assignment WELL BEFORE it is due !

- You WILL lose marks on this assignment if any of your files are missing. So, make sure that you hand in the correct files and version of your assignment. You will also lose marks if your code is not written neatly with proper indentation. See examples in the notes for proper style.
