Before you begin: In this graded lab you will be using <u>Observer</u> and <u>Template</u> patterns. First, examine the whole question, and then decide the pattern participants before you start your implementation.

In the post-apocalyptic world of "Fallout", your character, a skilled vault-dweller, traverses the harsh wastelands to gather valuable data from remnants of the old world to aid in survival and technological advancement as a quest. Your character should collect data from two devices named **WeatherInfo** and **WaterInfo.** You must tell **WeatherInfo** and **WaterInfo** devices to generate data.

Part 1 –Template Method: **WeatherInfo** and **WaterInfo** (30 pts)

1. Anyone who wants to collect data from previously mentioned devices should follow these steps in given order:

<u>**WeatherInfo**</u>   <u>**WaterInfo**</u>
generateData()   generateData()
openHatch()   openHatch()
read()   dipInWater()
                         read()

Note:

openHatch()	Prints a class specific string about the task the method does.
generateData()	generateData method sets an int attribute, "wData", randomly between 1-1000 for Weather and a double, "waData", between 0.0-1.0 for Water.
dipInWater()	Prints a class specific string about the task the method does.
read()	Indicates the data collection is completed.

.
2. Implement this utilizing a Template method and test it in your main.

Part 2 – Observer Pattern: **VaultComputer** (40 pts)

1. The **VaultComputer** observes these two machines.
2. **VaultComputer** state contains two attributes one for water and one for weather data.
3. Each device sends a notification after data collection is completed and the **VaultComputer** state is updated using the information obtained from **WeatherInfo** and **WaterInfo** devices.
4. Test your work in your main.

Part 3 – Third Party: Player (30 pts)

We would like a third party trigger the update instead of the devices. The third party here would be a "Player".
1. Player is a separate class that has "device" and "computer" attributes that are set via the constructor.
2. The player collects the data with a "collect ()" method. <u>After the data collection from these two devices are both completed,</u> the "Update" operation will be triggered. .
3. Modify your main accordingly so that it uses "Player" to finish the quest.

SUBMIT ONE/1/SINGLE JAVA FILE.
WRITE YOUR NAME and NUMBER IN THE SOURCE CODE