In this assignment, we will be implementing a container for meteor strike data. This container will use the Strategy pattern for its print output functionality. PLEASE provide a good example of your output in a separate file. Also, do submit all of your source code. Everything should be in a single .zip file.

1) Create an immutable class to hold a record (a row of data below is a record) called MeteorStrike of the following meteor strike data in Florida, (incidentally, data is real except for {month, date, hour, minute}. I made those up.)

| Name | Year | Month | Date | Hour | Minute | Location | Type | Weight (kg) |
|---|---|---|---|---|---|---|---|---|
| Bonita Springs | 1938 | 6 | 11 | 1 | 10 | Florida, USA | H5 | 41.8 |
| Eustis | 1918 | 1 | 2 | 2 | 11 | Florida, USA | H4 | 0.502 |
| Grayton | 1983 | 2 | 1 | 15 | 12 | Florida, USA | H5 | 11.3 |
| Okechobee | 1916 | 11 | 6 | 16 | 13 | Florida, USA | L4 | 1 |
| Orlando | 2004 | 12 | 31 | 17 | 14 | Florida, USA | Eucrite | 0.18 |

MeteorStrike should have a single constructor. E.g., that populates a single record from the data above.

MeteorStrike should use the GregorianCalendar class as the date and time container. However, for your output, please do not use any of GregorianCalendar methods. For this assignment, I want you to write your own using get() method. In particular, please use the following, .get(x), where x is replaced by Calendar.YEAR, Calendar.MONTH, Calendar.DATE, Calendar.HOUR or Calendar.MINUTE. Note, .get(Calendar.MONTH) returns an integer [0 … 11], so you would add 1 to this function's output. All of the other .get() method types above return the actual number you would expect.

2) To use the Strategy design pattern, one would need to define an interface for various print behaviors. Then, you would need to provide classes that implement this interface for each type of the behavior. Create one each for US, France, and Germany.

3) Create a class called MeteorStrikeCollection that is a container of MeteorStrikes. This class should either inherit from or contain as an instance variable ArrayList<>. This is your design choice.

4) Note that output generation starts at the MeteorStrike level while you are using a MeteorStrikeCollection as the container. I want it to be your design choice in how to implement the instance of the interface that is required by the Strategy design pattern. (There are a few forks in the road. There is probably no right or wrong here. Just think about what you need to do.)

5) In the main, instantiate MeteorStrikeCollection and add the five records above to it. There should only be one single method call in the main to switch between different types of output.

First print out all records using the USA way. Then print the last two records the French way. Finally print the second record the German way. **Dates, decimals, and weight measures** should be as below.

Decimals: USA  123.456; France 123,456; Germany 123,456
Dates: USA  12/31/2004; France 31/12/2004; Germany 2004-12-31
Weight: USA 3.307 lb; France 1,5 kg; Germany 1,5 kg     (Note, 1 kg = 2.20462 lb via the US convention)
Time: USA 2:10pm; France 14:10; Germany 14:10