City Data record system

Assignment brief:

Time 3 hours

You have been given some code that is currently being developed as part of a **City Data** record application. Not all the requirements have been implemented. It is your task to implement these and raise the coding standards of all the code.

Create a **project solution** (named **<Your Name><Student Number>Supplementary e.g. McGowanAidan3048614Supplementary).** Create a package named `supplementary`. Add **StartApp.java** to the solution and the **city_data.csv. Ensure your name and student number are placed in the Javadoc comments of all the classes you create.** The StartApp has been partially written with a menu.

The application will run (start) from the **StartApp.java**, initially reading the data from the **city_data.csv** file and then performing a number of menu-driven operations. The data in the csv relates to European Capital Cities. The provided data is representative only, and may not reflect real world statistics.

Part 1 – Data mapping, storage, Unit Testing and reading from file - 60%

Using your knowledge of OOP you should add/update the code based on the following:

1. Analyse the data in the **city_data.csv** and create a class (**City.java**). The file contains data for each of the cities. Validation / business rules are required as listed in Table 1 below.

Data field	Validation
City name	>0 to <=50 characters
Country	>0 to <=40 characters
Population	>0 to <=10000000 (10 million)
EU Status	Allowable values : EU or NON_EU
Number of Crimes	>=0 to <=100000
Time Zone	Allowable values : GMT, CET, EET, WET

Table 1 - Validation rules

2. Create a JUnit test class to fully test the **City** class.

3. In the **StartApp.java** class read and store the data (**city_data.csv**) in an appropriate JCF container (**cities**). Some of the fields for some records may be empty, null, not have the expected data types or violate the validation rules (Table 1). These records should **not** be included in the list of **cities** in the JCF container.

   Data mapping / parsing - when reading the data for the **EU Status** field you should map as follows:

Data field read (raw CSV input)	Map to :
0	NON_EU
1	EU

   An initial method stub for the read method has already been included in the StartApp.java file and requires completion.

[CONTINUED OVER]

Part 2 – Functions – 40%

Having read the data from the csv file complete the menu-driven functions as outlined below. An example of the expected format is shown for each function.

```
1. Display all cities
2. Display all EU Cities (in descending order by city name)
3. Display City Name and crime rate per 1000 population
4. Display City with highest population
5. Write to file all cities in EET timezone (new thread needed)
6. Quit
Enter option ...
```

Note after each option 1-5 is complete the menu should be redisplayed on screen. Option 6 should end the application.

1. **Display all Cities data**. Number each entry. Example output…

```
Enter option ...
1

1)
City [cityName=London, country=United Kingdom, population=8982000,
euStatus=NON_EU, numberOfCrimes=56432, timeZone=GMT]

2)
City [cityName=Paris, country=France, population=2140526, euStatus=EU,
numberOfCrimes=42187, timeZone=CET]
etc…
```

2. **Display all EU Cities** (sorted by City Name descending). Same output formatting as before. Example output…

```
Enter option ...
2

1)
City [cityName=Zagreb, country=Croatia, population=173286, euStatus=EU,
numberOfCrimes=18457, timeZone=CET]

2)
City [cityName=Warsaw, country=Poland, population=1790658, euStatus=EU,
numberOfCrimes=12457, timeZone=CET]

3)
City [cityName=Vienna, country=Austria, population=1897491, euStatus=EU,
numberOfCrimes=15872, timeZone=CET]
etc…
```

[CONTINUED OVER]

3. **Display City Name and crime rate per 1000 population** . Example output…

```
Enter option ...3

City            Crime rate (per 1000 of population)
London          :  6.28
Paris           : 19.71
Berlin          :  5.00
Madrid          :  9.00
```

The Crime rate for each city is calculated by dividing the **number of crimes** by the **population** divided by **1000**. Display to two decimal points.

$$\text{Crime rate} = \frac{\text{Number of crimes}}{\text{Population / 1000}}$$

Display to two decimal points.

4. Display City with highest population

```
Enter option ...
4

Highest population city
 City [cityName=London, country=United Kingdom, population=8982000,
euStatus=NON_EU, numberOfCrimes=56432, timeZone=GMT]
```

5. Write to file all cities in **EET** timezone (new thread needed)

Create new Thread to write all Cities in the EET Timezone to a file named **TimeZoneCities.csv**. The contents of the file should look this :

```
City_name,Country,Population
ATHENS,Greece,664046
HELSINKI,Finland,648042
```

Note the inclusion of header and the capitalisation of the City names.

When complete, compress (zip) the entire ***Eclipse solution*** and upload to **Assignments** (Supplementary) on CANVAS.

Now : check the uploads to ensure you have submitted the correct files (in the correct area).

[END]