

# Assignment 8

## Radon Gas (Lists, File I/O, and Classes)

### Radon Level Analysis Application

#### Background:

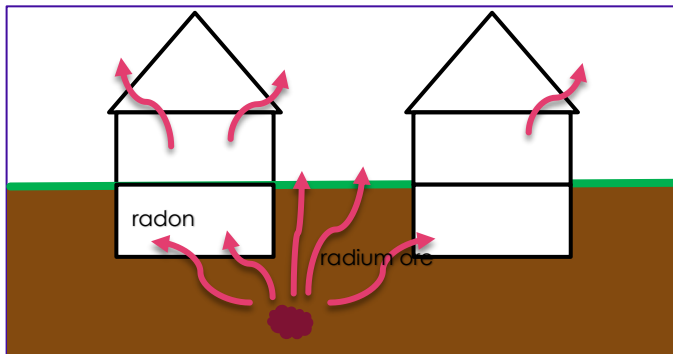
Radon  $^{86}\text{Rn}$  is a radioactive, colorless, odorless, tasteless noble gas. Prolonged exposure to the gas can lead to health issues: most commonly non-smoking related lung cancers. It is the second leading cause of lung cancer.

Radon is most highly associated with naturally occurring radium ores which are widely distributed across Canada (see map).

Though some areas are more likely to have high levels of radon, all dwellings should be tested, as local concentrations can vary dramatically from house to house.

Because radon is very dense, it tends to collect in basements so that is where testing is done.

Health Canada recommends that houses that **exceed an average annual radon concentration of 200 becquerels per cubic metre ( $200 \text{ Bq/m}^3/\text{y}$ )** should have the basement walls and floors sealed and additional forced air systems installed to vent the radon<sup>1</sup>.

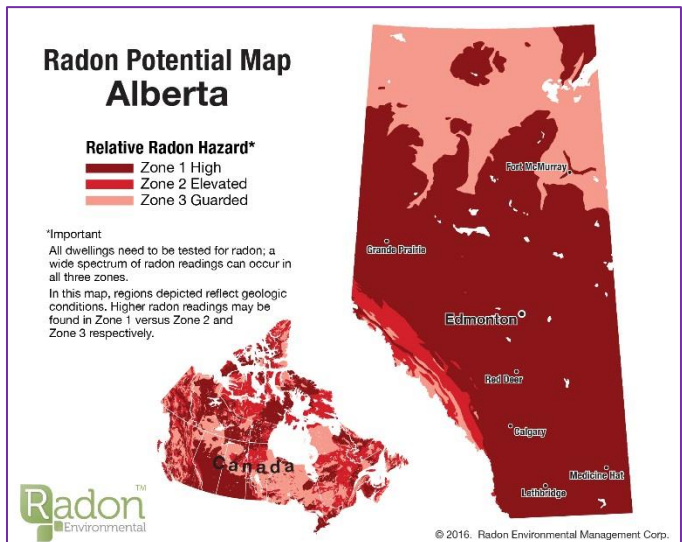


#### Problem:

Assuming that there are separate single family homes on one side of a single street, Rue Marie Curie, by convention the address numbers of adjacent dwellings will differ by 4. Thus you will find 1504 Rue Marie Curie next to 1508 Rue Marie Curie.

One can also assume that each dwelling has an associated basement volume (between 100 and 500  $\text{m}^3$ ) and an average daily radon level reading (between 0.00000 and 0.01500 Bq per day).

The problem then is to create syntactically correct and runnable Java code that will process that data.



<sup>1</sup> [http://www.hc-sc.gc.ca/ewh-semt/radiation/radon/faq\\_fq-eng.php#indoor](http://www.hc-sc.gc.ca/ewh-semt/radiation/radon/faq_fq-eng.php#indoor)

To do this, you would need to create the following:

1. A **House** class that contains information about the level of Radon in the basement. This class will have the following as instance variables:
  - a. The house number
  - b. The street name
  - c. The basement volume
  - d. The average daily Radon reading

In addition, you will need to be able to calculate the yearly Radon level for the house. This yearly level is the product of the average daily Radon reading, the basement volume and the number of days in a standard year. **This yearly level should not be an instance variable, but should instead be calculated when required!** You will also need to be able to print the information about each house when required.

2. A **Radon Processor** class that processes the data. This class will have at least one ArrayList which contains the houses in the block. Other ArrayLists may be required and should be included whenever necessary. This class should have methods that do the following:
  - a. Read data from a text file to instantiate **House** objects and populate the ArrayList of houses. This is to be done as the application is started. The name of the data file is to be inputted by the user.
  - b. Print the contents of the ArrayList. The house number and street name would be sufficient. This will be used to confirm that the processing was done properly.
  - c. Print the details for each house. This includes all instance variable values as well as the calculated yearly Radon level.
  - d. Search the ArrayList for a particular address (for this assignment, we will only use the house number as a search criteria). This will return the index if found or -1 if not in the ArrayList. This will be used by other methods to confirm the existence of this address before any processing can be done.
  - e. "Demolish" a house. A search must be done to confirm that the house exists. If so, then the house is to be removed from the ArrayList. Otherwise, a message that no house was demolished is printed. You can confirm that this was done properly by printing the ArrayList (as a separate step).
  - f. Add an infill to a vacant lot. First, confirm that the address does not exist in the ArrayList. Hence, the lot that is allocated to this address is indeed vacant. An infill is a tall thin duplex. Hence, you will need to instantiate 2 house objects and insert into the ArrayList in the proper location. (Note: you may need to create an alternate search method for this). For the daily Radon reading, you should take the average of the readings of the houses on either side of the lot. If the lot is at the beginning of the block or at the end of the block, assume the value will be the same as the lot next to it. For the address, the first one will have the regular house number, and the second one will have a number that is 2 away. For example, you create a duplex in the lot allocated for house number 1504. The first of the duplex will have a house number 1504, the second one, 1506.

- g. Create a report that outputs the address, calculated Radon level, and relative danger rating for each house. Each house should be rated as to the relative danger of Radon exposure according to the following criteria:
- $< 200 \text{ Bq/m}^3/\text{y}$  is considered safe
  - $< 800 \text{ Bq/m}^3/\text{y}$  is considered hazardous with remediation recommended
  - $800 \text{ Bq/m}^3/\text{y}$  or more is considered dangerous with remediation required immediately

The report should look like:

House Address	Level	Rating
1400 Rue Marie Curie	132.0056	Safe – no remediation required
1404 Rue Marie Curie	456.7894	Hazardous – remediation required
1412 Rue Marie Curie	965.0003	Dangerous – immediate remediation required

- h. Write the information contained in the ArrayList of houses into a text file in the same format as the original text file. In this way, this output file can be used as an input at a later date. This is to be done automatically before exiting the application.

For the **RadonProcessor** class, the methods that will process the menu choices should be declared as public. Other methods (helper methods) should be declared as private. In addition, describe what each method does with a one to two line description in addition to the documentation in the method body.

3. A **RadonClientApplication** class that contains the public `run()` and a variety of private helper methods. The application will present a menu to the user and process the user's choice by calling the appropriate methods of the processor class. A skeleton of this application class will be provided.

In addition, all inputs via keyboard should be in client application only - **do NOT do any keyboard input in RadonProcessor.**

As with the previous assignments, a base solution will be provided. In this base solution, a no source **House** class, with full JavaDoc, will be included. You may use this class to work on the assignment without having to develop the class. When you are satisfied that you have met the requirements of the assignment, you can develop this class and use your developed class to test your work. You will also have a sample data that you can use to develop your work.