Week 12

Reading/Writing Files

1. Describe how to set up a `Scanner` for reading text files.
2. When you have finished editing the text file, what must you not forget to do?
3. When opening a file, which exception has a chance to be thrown?
4. Write a program that requests the user to enter a sentence, then output a text file called "awesome.txt" which contains the input sentence.
   Below is an example scenario:

   ```
   Please enter a sentence:
   Rayquaza wears lipstick!
   ```

   A text file called "awesome.txt" should be created and contain the above sentence.
5. Describe how to *append data* as opposed to overwrite data when using a `PrintWriter` object.

Processing Text

1. What is the purpose of changing a `Scanner` object's delimiter to another String *besides white space*?
2. List five useful methods, in your opinion, for manipulating Strings.
3. List five useful methods, in your opinion, for working with characters. (The `Character` class is handy)
4. Consider the following text:

   ```
   Brock Rock
   Misty Water
   Lt.Surge Electric
   Erika Grass
   Koga Poison
   Sabrina Psychic
   Blaine Fire
   Giovanni Ground
   ```

   Write a program that prints the above text in a more descriptive format. The file is named "data.txt" and is located in the root directory. Below is the expected output:

   ```
   Brock is the Rock-type Gym Leader.
   Misty is the Water-type Gym Leader.
   Lt.Surge is the Electric-type Gym Leader.
   Erika is the Grass-type Gym Leader.
   Koga is the Poison-type Gym Leader.
   Sabrina is the Psychic-type Gym Leader.
   Blaine is the Fire-type Gym Leader.
   Giovanni is the Ground-type Gym Leader.
   ```

Command-line Arguments

1. Describe the syntax for passing arguments through the command-line.
2. Where are the command-line arguments stored in a program?
3. Describe how to access passed command-line arguments.

Exceptions and Exception Handling

1. What is the purpose of an `Exception`?
2. Describe the syntax of a `try-catch` block.
3. Describe the syntax of a `throws` statement.
4. Consider the following code:

```
public class Glitch {
    public static void main(String[] args){
        int pokedexNo = 0;
        if(pokedexNo > 0){
            System.out.println("Wild Pokemon!");
        } else {
            throw new Exception("Missingno!");
        }
    }
}
```

   Write the above code again after modification in the following two scenarios:

   (a) Handle the `Exception` with a `try-catch`

   (b) Handle the `Exception` with a `throws`

5. List five common predefined `Exception` objects you have come across. (E.g. `NullPointerException`)
6. What happens if an `Exception` is not caught and the program is executed?
7. Explain why more specific `Exception` objects should be caught before more general ones.