Homework #8 – Streams

In this assignment you are asked to implement a program that reads a list of `Person` objects from a file, outputs those objects to an in-memory stream, then outputs the contents of the in-memory stream to `stdout`. The following UML class diagram shows the class that will be read from streams and written to streams.

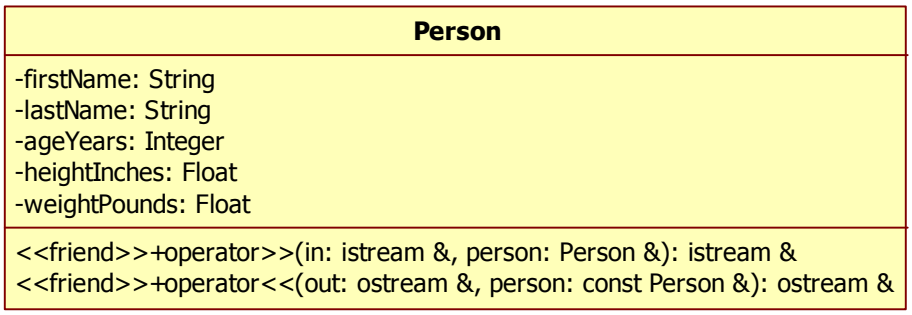

Figure 1. UML class diagram for class Person

1. **(1 point)** Meet these basic requirements:
   a. All non-test code must be implemented in a namespace based on your first and last name (e.g. "`RayMitchell`").
   b. The `Person` class must be defined in a file named "`Person.h`"; all member functions must be defined in a file named "`Person.cpp`".
   c. The `main` function and tests must be placed in a file named "`hw8.cpp`".
   d. Make sure `const` is used correctly everywhere within the assignment. Be sure to check all pointer parameters, reference parameters, and member functions for proper "const-ness".

2. **(3 points)** Define class `Person` exactly as shown in the UML diagram. *Do not add any members not shown in the diagram*.

   The stream insertion operator (`operator<<`) should output the attributes of the `Person` in the following format:

   ```
   firstName lastName ageYears heightInches weightPounds
   ```

   Here is an example of the output expected from the stream insertion operator:

   ```
   John Smith 37 72.25 189.37
   ```

   The stream extraction operator (`operator>>`) should extract a `Person` in the same format.

3. **(5 points)** Write a test program that implements the following pseudocode:

   ```
   Open a file named "hw8-input.txt" using an fstream
   If the file opened successfully:
   ```

```
While the end of the file has not been reached:
     Extract the next person from the file
     Insert the person and a newline into a stringstream
Output the contents of the stringstream to stdout
```

Demonstrate your program working with file "`hw8-input.txt`" containing the following:

```
John Smith 37 72.25 189.37
Sally Jones 26 60.18 107.88
Aaron Anderson 12 55.50 89.30
Janet Anderson 10 49.00 60.98
Brian Doe 28 76.11 238.00
Brett Daniels 48 69.01 189.74
```

4. **(1 point)** Make sure your source code is well-commented, consistently formatted, uses no magic numbers/values, follows a consistent style, and is ANSI-compliant.

Place all source code and a screen capture of the output produced by your program in a single Word or PDF document. Submit this document.