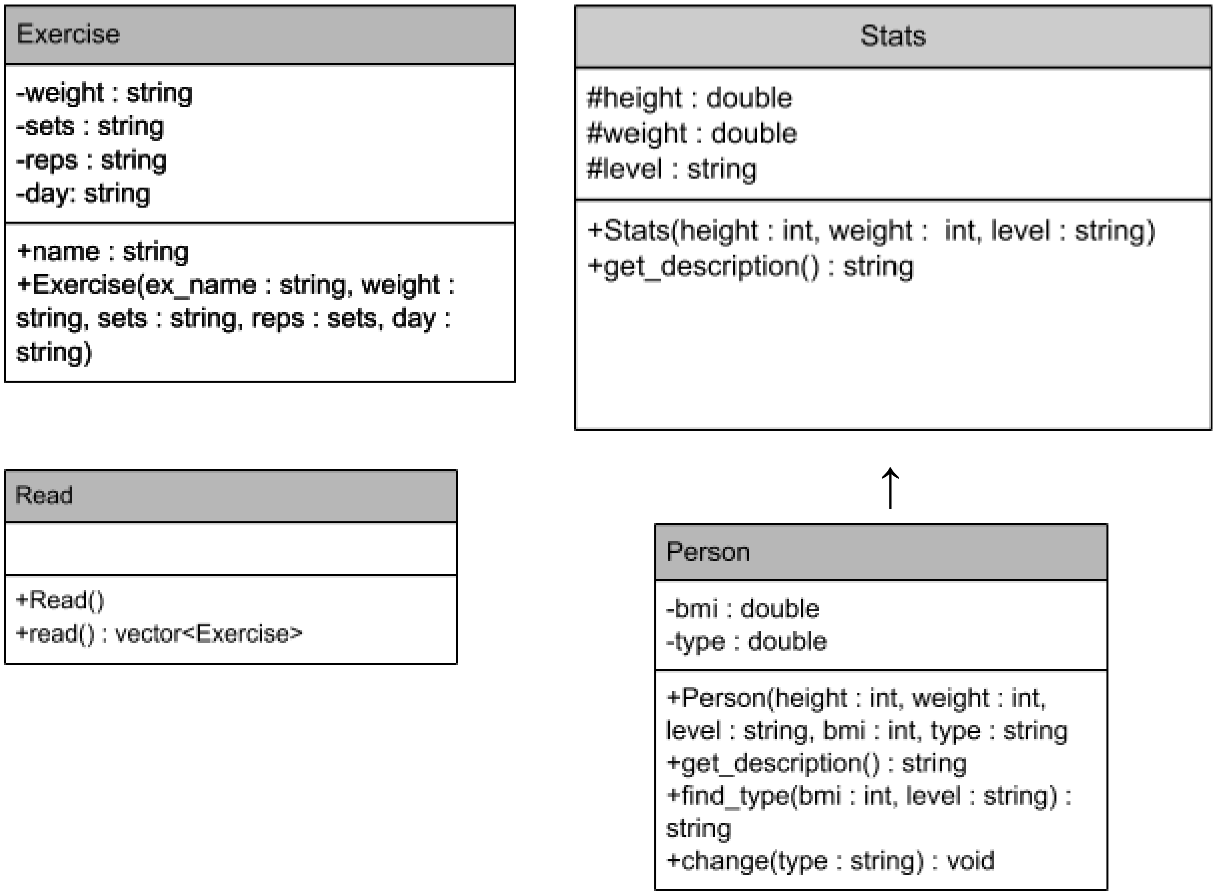
Erwin Goscin
Flex Project Submission 1

## UML Diagram


Exercise
-weight : string
-sets : string
-reps : string
-day: string
+name : string
+Exercise(ex_name : string, weight : string, sets : string, reps : sets, day : string)
Stats
#height : double
#weight : double
#level : string
+Stats(height : int, weight : int, level : string)
+get_description() : string
Read
+Read()
+read() : vector<Exercise>
Person
-bmi : double
-type : double
+Person(height : int, weight : int, level : string, bmi : int, type : string
+get_description() : string
+find_type(bmi : int, level : string) : string
+change(type : string) : void

## Description

This program will create a workout routine for the user. There will be menu to edit routines for each day of the week which are read from and stored in a plan.txt file. The user may add exercises for each daily routine by inputting the name, sets, reps, and weight for a particular exercise. They will also be able to input their body statistics and have the program calculate their Body Mass Index or BMI *([weight in lbs ÷ (height in inches)^2] x 703)* and activity level(low or high) and adjust how many sets should be performed based on these results. Exercises may also be deleted from the routine/plan.

The Person class will **inherit** the Stats class **(**Person.h line 17, Person.cpp line 23). It will use **polymorphism(**Stats.h line25, Stats.cpp line 21, Person.h line 25, Person.cpp line 28) for the **get_description()** function. This way when the **show_stats()** function (main.cpp lines 85, 132) is called using a const Stats& s object as an argument, it will use the **get_description()** function from the Person class instead of the Stats class and return a string containing the description of the user. It will display height, weight and BMI along with the type(normal, athlete, sedentary). A **find_type()** **<u>static</u>** function (Person.h line 26, Person.cpp line 37) will determine whether the person fits into 3 categories: Normal, Athlete, Sedentary and storing it into a variable called type. This will be done by comparing the BMI (normal<=24, high>24) and activity level(low or high). It will take bmi and level variables as arguments and return a string of Normal, Athlete, or Overweight.

- If bmi normal & activity low, type = *normal*
- If bmi normal & activity high, type = *athlete*
- If bmi high & activity low, type = *sedentary*
- If bmi high & activity high, type = *normal*

If *normal*, adjust all routines to 4 sets and 8 reps for every exercise.
If *athlete*, adjust all routines to 5 sets and 6 reps for every exercise.
If *sedentary*, adjust all routines to 6 sets and 15 reps for every exercise.

There will be a Read class to read the contents of the plan.txt file which can then be stored in a vector of exercise objects called exercise_list. It will use the **<u>static</u>** function **read()** (Read.h line 22, Read.cpp line 25) and will return a vector of exercise objects;

These exercises will be adjusted using a **<u>static</u>** void function called **change()** (Person.h line 27, Person.cpp line 52) that takes the string variable type as an argument. It will access the sets and reps variables of the exercise_list vector and change them accordingly.

There will be an Exercise class object to store each exercise with its information. Private data members will be used for the exercise information along with setter and getter functions to demonstrate **<u>encapsulation</u> (**Exercise.h, Exercise.cpp lines 38 & 57).

It will have two **<u>static</u>** functions (Exercise.h lines 33,34, Exercise.cpp lines 29,67) void **add()**, and void **remove()** to add and remove exercises from the exercise_list vector which is the written to the plan.txt file. Also, the following **<u>friend overload operator</u>** function (Exercise.h line 47) will be used to output the routine:

**friend std::ostream& operator<<(std::ostream& out, const std::vector<Exercise>& exercise_list).**

I think this program might be eligible for EC1 since others could use it to create a workout routine and calculate BMI and fitness level.