Due date: Today, at the end of the lab period

1) Define a class called **`Player`** that implements `Comparable <Player>` with health and name attributes and has getter/setter methods to retrieve/access them. Define the appropriate constructors and give it a useful `toString()` method.
2) Define a class called `LinkedList` with an inner nested class `Node`.
3) The `Node` class has private attributes (`Player player; Node next;`). Define the appropriate constructor and `toString()` methods.
4) Class `LinkedList` has two attributes (`Node head` and `int size`).
5) Define the following `LinkedList` methods:
   `insert (Player p); remove (Player p); toString ();`
6) NOTE: your `insert()` method must be ordered. This means that your newly added `Node` should be placed in an ordered fashion such that your resulting linked list is ALWAYS sorted by the `Player's health`.
   For example:
   - If `Player p` has `health 2000`, then `insert()` will produce the following linked list: `p→null`
   - If `Player p2` has `health 1000`, then `insert()` will produce the following linked list: `p2→p→null`
   - If `Player p3` has `health 500`, then `insert()` will produce the following linked list: `p3→p2→p→null`
   - If `Player p4` has `health 5000`, then `insert()` will produce the following linked list: `p3→p2→p→p4→null`
7) NOTE2: YOU SHOULD ALWAYS MAINTAIN ENCAPSULATION! Do NOT insert the Player argument, insert a copy/clone of the Player.
8) Create a Driver program to test your application.

Submission: Once you are done, upload your program to the https://fis.encs.concordia.ca/eas/
Please name your file following this convention: *lab11_studentID*, where *studentID* is your Student ID number.