Linked List Project. Roster

1. A wrestler object consists of a name (String) and a weight(int).

Complete the wrestler class. Include all necessary constructors, setters. Getters, and a toString(). Copy that code below.

1. A roster is a collection of Wrestlers using a ListNode. Complete the following methods. Include a tester to test your methods

public class Roster

{

// pointer to linked list of wrestlers on the roster sorted

// by weight (lighest to heaviest)

private ListNode firstWrestler;

public Roster()

{

firstWrestler=null;

}

/\*\*

\* precondition: the teamList field is null or it points to the

\* first node of a linked list

\*

\* postcondition: w has been added to the list in sorted order (by weight)

\*/

public void addWrestler(Wrestler w)

{

}

/\*\*

\* precondition: the teamList field is null or it points to the

\* first node of a linked list

\*

\* return a String representation of the roster

\*/

public String toString ()

{

}

/\*\*

\* precondition: the teamList field is null, or it points to the first node of a linked list

\* **the list is sorted according to the weight**

\* postcondition: returns the pointer to the node in the linked list for the wrestler \*whose weight is closest to the given weight without going over. Returns null if there \*are no wrestlers or none whose weight is less than or equal to the given weight

\*/

public ListNode wrestlerBefore( int weight)

{

}

/\*\*

\* Displays wrestlers less than weight

\*/

public void SmallerList(int weight)

{

}