

CS56—Final Exam E03

E03, W16, Phill Conrad, UC Santa Barbara 03/16/2016

HAND THIS IN WITH YOUR EXAM.

YOU MAY USE THIS FOR SCRATCH WORK, BUT ALL ANSWERS SHOULD BE ON YOUR EXAM PAPER.

Name: _____

Umail Address: _____@ uemail.ucsb.edu

Description of TeamPlayer class

The class `edu.ucsb.cs56.w16.e03.TeamPlayer` is public and represents individuals that are part of a sports team. The class has four private attributes:

- `private String fname`—first (given) name
- `private String lname`—last (family) name
- `private int jerseyNumber`—the jersey number of the player, i.e the number the player wears on
- `private String position`—this is a string describing the position the player has, and differs from sport to sport. For example: goalie, linebacker, quarterback, guard, etc.

There are public getters and setters for each of these attributes. The getter and setter for `fname` is shown below. The remainder of the setters and getters follow this convention.

- `public String getFname()`
- `public void setFname(String fname)`
- etc.

There is also a single constructor that takes the parameters shown here:

```
public TeamPlayer(String fname, String lname,  
                  int jerseyNumber, String position)
```

Javadoc for `java.util.ArrayList` is on the other side

java.util.ArrayList<E>

boolean	add(E e) Appends the specified element to the end of this list.
void	add(int index, E element) Inserts the specified element at the specified position in this list.
boolean	addAll(Collection<? extends E> c) Appends all of the elements in the specified collection to the end of this list, in the order that they are returned by the specified collection's Iterator.
boolean	addAll(int index, Collection<? extends E> c) Inserts all of the elements in the specified collection into this list, starting at the specified position.
void	clear() Removes all of the elements from this list.
Object	clone() Returns a shallow copy of this ArrayList instance.
boolean	contains(Object o) Returns true if this list contains the specified element.
void	ensureCapacity(int minCapacity) Increases the capacity of this ArrayList instance, if necessary, to ensure that it can hold at least the number of elements specified by the minimum capacity argument.
E	get(int index) Returns the element at the specified position in this list.
int	indexOf(Object o) Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.
boolean	isEmpty() Returns true if this list contains no elements.
Iterator<E>	iterator() Returns an iterator over the elements in this list in proper sequence.
int	lastIndexOf(Object o) Returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element.
ListIterator<E>	listIterator() Returns a list iterator over the elements in this list (in proper sequence).
ListIterator<E>	listIterator(int index) Returns a list iterator over the elements in this list (in proper sequence), starting at the specified position in the list.
E	remove(int index) Removes the element at the specified position in this list.
boolean	remove(Object o) Removes the first occurrence of the specified element from this list, if it is present.
boolean	removeAll(Collection<?> c) Removes from this list all of its elements that are contained in the specified collection.
protected void	removeRange(int fromIndex, int toIndex) Removes from this list all of the elements whose index is between fromIndex, inclusive, and toIndex, exclusive.
boolean	retainAll(Collection<?> c) Retains only the elements in this list that are contained in the specified collection.
E	set(int index, E element) Replaces the element at the specified position in this list with the specified element.
int	size() Returns the number of elements in this list.
List<E>	subList(int fromIndex, int toIndex) Returns a view of the portion of this list between the specified fromIndex, inclusive, and toIndex, exclusive.
Object[]	toArray() Returns an array containing all of the elements in this list in proper sequence (from first to last element).
<T> T[]	toArray(T[] a) Returns an array containing all of the elements in this list in proper sequence (from first to last element); the runtime type of the returned array is that of the specified array.