Handout C

`Student.java` for questions 6,7,8 (pages 4,5,6)

A partial listing of the code for `Student.java` appears below.

Note that the code is incomplete (see the comments at lines 22-24).

During the exam, you'll be asked to supply some of the missing code.

```
 1 // Student.java
 2 import java.util.ArrayList;
 3 import java.util.Comparator;
 4
 5 public class Student implements Comparable<Student> {
 6
 7     public Student(int perm, String name, double gpa, String major) {
 8         this.perm = perm; this.name=name;
 9         this.gpa = gpa; this.major=major;
10     }
11
12     private String name;
13     private int perm;
14     private double gpa;
15     private String major;
16
17     public String getName() {return this.name; }
18     public int getPerm()    {return this.perm; }
19     public double getGpa()  {return this.gpa;  }
20     public String getMajor()  {return this.major;  }
21
22     @Override
23     public String toString() {
24            String result =  "Student(" + perm;
25            result += ",\"" + name + "\",";
26            result += String.format( "%.2f", gpa) + "," + major + ")";
27            return result;
28     }
29
30     //  ***************************************
31     //  * MORE METHODS WOULD BE DEFINED HERE   *
32     //  ***************************************
33 }
```

1

Handout C for e02 CS56 F17

2

Handout C for e02 CS56 F17

class `java.util.ArrayList<E>`

The following excerpts from the javadoc for `java.util.ArrayList<E>` may be helpful to you in completing this exam.

Inheritance Hierarchy (complete)

```
java.lang.Object
  java.util.AbstractCollection<E>
    java.util.AbstractList<E>
      java.util.ArrayList<E>
```

All Implemented Interfaces:	`Serializable, Cloneable, Iterable<E>, Collection<E>, List<E>, RandomAccess`
Direct Known Subclasses:	`AttributeList, RoleList, RoleUnresolvedList`

Constructors (complete)

`ArrayList()`	Constructs an empty list with an initial capacity of ten.
`ArrayList(Collection<? extends E> c)`	Constructs a list containing the elements of the specified collection, in the order they are returned by the collection's iterator.
`ArrayList(int initialCapacity)`	Constructs an empty list with the specified initial capacity.

Most important methods, with brief description

`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.
`void`	`clear()`	Removes all of the elements from this list.
`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.
`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.
`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.
`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.
`void`	`sort(Comparator<? super E> c)`	Sorts this list according to the order induced by the specified `Comparator`.

Additional methods, listed by method signature only.

`boolean addAll(Collection<? extends E> c)`	`boolean addAll(int index, Collection<? extends E> c)`
`Object clone()`	`boolean contains(Object o)`
`void ensureCapacity(int minCapacity)`	`void forEach(Consumer<? super E> action)`
`Iterator<E> iterator()`	`ListIterator<E> listIterator()`
`ListIterator<E> listIterator(int index)`	`boolean removeAll(Collection<?> c)`
`boolean removeIf(Predicate<? super E> filter)`	`protected void removeRange(int fromIndex, int toIndex)`
`void replaceAll(UnaryOperator<E> operator)`	`boolean retainAll(Collection<?> c)`
`Spliterator<E> spliterator()`	`List<E> subList(int fromIndex, int toIndex)`
`Object[] toArray()`	`<T> T[] toArray(T[] a)`
`void trimToSize()`	

Methods inherited from:

`class java.util.AbstractList`	`equals, hashCode`
`class java.util.AbstractCollection`	`containsAll, toString`
`class java.lang.Object`	`finalize, getClass, notify, notifyAll, wait, wait, wait`
`interface java.util.List`	`containsAll, equals, hashCode`
`interface java.util.Collection`	`parallelStream, stream`

End of Handout