Sorting

Programming Projects:

1) Person class

- Start Eclipse and create a new Java project called ‘lab7’

- Create a class called **Person**. Add attributes that store the person’s height (in cm) the person’s age (in years) and the person’s gender (as a boolean). Also add a constructor, and getters and setters for each attribute. Also add a **toString**() method which displays the person’s details.

- Create another class called **Census** which maintains a list of people (Person instances). The people should be stored using an array list, i.e.

  ```
  private List<Person> people = new ArrayList<Person>();
  ```

- Add the following methods to the **Census** class along with their implementations -

  ```
  void addPerson(Person person);
  boolean removePerson(Person person);
  int countPeople();
  int countMales();
  int countFemales();
  ```

- Add a **toString**() method which shows the details of all people in the census.

- Create a **Driver** class which creates an instance of **Census** and makes calls to add and remove several **Person** instances, and to test the other methods.

Think you’re finished? Check you have commented your code and added Javadoc to the methods and classes.

2) **Sorting** Methods

- Add the following methods to the Census class.

```
void sortOnAge();
void sortOnHeight();
void sortOnGender();
```

- Implement each of these methods using the **Collections.sort** method. Implement an anonymous inner class which sorts the people using the appropriate attribute for each method e.g.

```
Collections.sort(people, new Comparator<Person>() {

	@Override
	public int compare(Person p1, Person p2) {
		// TODO compare p1 to p2 here using appropriate attribute
	}}
);
```

- Amend the **Driver** class so that is sorts the Census data using each of these methods, then prints out the list to show the correct order.

Think you're finished? Check you have commented your code and added Javadoc to the methods and classes.