Test Appendix:

```
/**
 * Method to get the closest agent within the passed distance
 * to this agent but NOT this agent.
 * We'll search the input list of the kind
 * of objects to compare to.
 * @param distance the distance to look within
 * @param list the list of agents to look at
 * @return the closest agent in the given distance or null
 */
public Agent getClosest(double distance,
                        LinkedList list) {

  // Keep track of closest so-far
  Agent closest = null;
  double closestDistance = 999;
  double currDistance = 0;
  Agent current = null;

  // loop through the linked list looking for the closest Agent
  for (int i = 0; i<list.size(); i++) {
    current = (Agent) list.get(i);
    if (current != this) {
      currDistance =
   current.getDistance(this.getXPos(),this.getYPos());
      if (currDistance < distance) {
        if (closest == null ||
            currDistance < closestDistance) {
          closest = current;
          closestDistance = currDistance;
        }
      }
    }// current != this
  }// for loop
  return closest;
}
```

Your Name:______________________

```
/**
   * How a Person acts
   */
  public void act() {

    // Is there a person within infection range of me?
    Simulation sim = getSimulation();
    PersonAgent closePerson =    (PersonAgent)
               getClosest(20, sim.getAgents());

    if (closePerson != null) {
      // If this person is infected, and I'm not infected
      if (closePerson.infected && !this.infected) {
        // I become infected
        this.infect();
      }
    }

    // Run the normal act() -- wander aimlessly
    super.act();
  }
```

```
 /**
   * Replace the one sound with the other sound
   * in all the elements from me on.
   * Two sounds are equal if come from same filename
   * @param oldSound sound to be replaced
   * @param newSound sound to put in its place
   */
  public void replace(Sound oldSound, Sound newSound) {
    if (this.mySound.getFileName() != null) {
      if (this.mySound.getFileName().equals(oldSound.getFileName()))
{
        this.mySound = newSound;
      }
    }

    if (next != null) {
      next.replace(oldSound,newSound);
    }
  }
```