```
class Room {

  ArrayList<NPC> occupants = new ArrayList<NPC>();
  int roomSize;//for implementing multiple rooms and constraining how many agents can be in a room at one time.
  int exits;
  boolean empty;
  String roomName;
  Room(String _roomName, int _roomSize, int _exits) {
    this.roomName = _roomName;
    this.roomSize = _roomSize;
    this.exits = _exits;
    this.occupants = null;
    this.empty = true;
  }

  void population() {
    println("Characters currently in " + roomName + " are:");
    for (int i = 0; i < occupants.size(); i++) {
      NPC currChar = occupants.get(i);
      println("roomSlot:" + i + " " + currChar.name + " " + currChar.roomNameIn);
    }
  }

  int numOfOccupants() {
    int noOfOcc = 0;
    NPC dummy;
    for(int i = 0; i < occupants.size(); i++){
      dummy = occupants.get(i);
      if(dummy != null){
       noOfOcc += 1;
      }
    }
    return noOfOcc;
  }

  void populate(NPC[] charID) {
    occupants = new ArrayList<NPC>(charID.length);
    for (int i = 0; i < charID.length; i++) {
```

```
   occupants.add(charID[i]);
   occupants.get(i).roomNameIn = roomName;
  }
 }

 void clearRoom(Room a) {
  if (a != null) {
   if (a.empty == true) {
    for (int i = 0; i < a.occupants.size(); i++) {
     a.occupants.remove(i);
    }
   } else {
    print("The Room is already empty");
   }
  } else {
   print("You need to initialise a room first");
  }
 }


}
```