Name (PRINTED): _____

Student ID #: _____

Section # (or TA's name and time): _____

CMSC 131 Quiz #9 Spring 2009

For both of the questions on the other side of this paper, assume the classes shown here:

```
public class Door {
     private int ht;
     private int rating;

     public Door(int ht, int rating){
        // valid constructor for the Door class
     }
     public Door(Door oldDoor){
        // valid copy constructor for the Door class
     }
     public boolean equals(Door rtDoor){
        // valid equals method for the Door class
     }
     //other methods appear here
}
-------------------------------------------
public class House {
     private Door[] all;
     private String address;

     public House(Door[] doorList, String address){
        //valid constructor for the House class
     }
     public House (House oldHouse){
        //valid copy consructor for the House class
     }
     public boolean equals(House otherHouse){
        //valid equals for the House class
     }
     //Other methods appear here
}
```

Turn to the other paper to answer the questions. You may separate the two pages (it may be easier for you that way), but make sure you put your name on both and make sure you turn both back in at the end of the quiz.

This page intentionally blank.

Name (PRINTED): _____

1. Assume, as shown on the other side that there is a class called `Door` that has a good copy constructor. You must assume that both the `Door` and the `House` classes are mutable (because you can't see what other methods are there). As shown, the `House` class has an instance variable of type `String` called `address`. It also has an instance variables of type `Door[]` called `all`.

   In the space below, write a good copy constructor for the `House` class. Since the `Door` objects are mutable, be sure that your copy constructor does not result in any aliasing of `Door` objects.

```
public House (House old){
//write the code needed for this method









}
```

2. Write the JUnit test that would determine if your House copy constructor (written for the other question) worked correctly. Unlike most JUnit tests, you only need to write enough code to test one case of the copy constructor for the House. Also, this test does not need to be complete. It only needs to make sure you have the new house with the same values. It does not need to make sure aliasing did not occur. (note: this restriction is only for the quiz not for real life.)

```
import junit.framework.TestCase;

public class HouseTest extends TestCase {

     public void testHouseHouse() {
        // Write the code that would need to be here to test
        // the copy constructor above.
        // Only one test is needed - not several as shown in class
        // or as you shouls have done on the project









     }
}
```