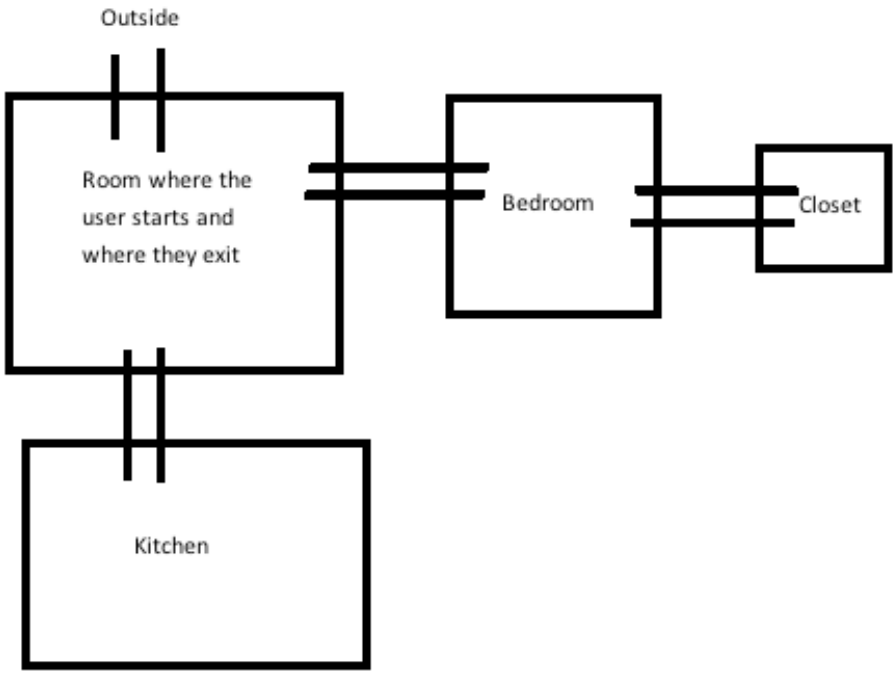
Assignment 7

CSC 202

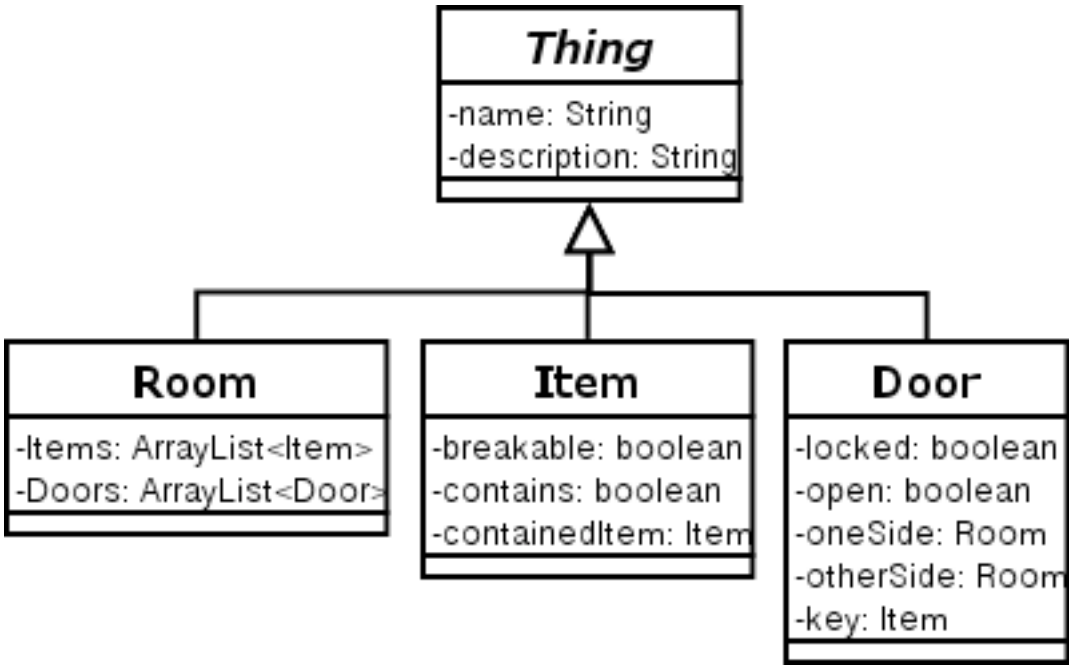
Objective: Gain greater understanding of objects, classes, and inheritance.

Instructions:

1. You are to build on the base of what you created in Project 6.
2. You are to create a text-based game that follows a basic storyline. The basic storyline is that the user gets lost in the woods at night, finds an old building then enters the building. Upon entering the building the door locks behind them. The user has to find their way out of the building by picking up items, solving puzzles, etc.
3. The following is the map of the building. There is the main room that connects to a bedroom and a kitchen. Besides connecting to the main room, the bedroom connects to a closet as well.



The following is a UML diagram for you to implement:



The entire game is text-based. The following are valid commands (and it is the help message):

help
look
look at *thing*
pick up *item*
use *item*
throw *item*
go through *door*
quit
introduction

1. If the user types something that is not valid then display the help message.
2. If the user types "help" then display the help message.
3. If the user types "look" then print the name and description of the following:
    1. current room
    2. doors in the room
    3. items in the room
    4. items in the user's backpack
4. If the user types "look at $thing$" then print the name and description of the $thing$ if it is in the room (or is the room).
5. If the user type "pick up $item$":
    1. It has to be an $item$ – can't pick up room objects or door objects.
    2. Parse out what the $item$ is.
    3. If it is in the room, then "pick it up" and put it in your "backpack."
    4. Your "backpack" contains all the items that you have picked up. There is no limit to how many items you have in the backpack.
6. If the user types "use $item$" then use the item. How? If the $item$ is a key that unlocks a door, then unlock that door and open it.
7. If the user types "throw $item$" then throw the item. If the $item$ is breakable then it breaks. If it contains something then that $item$ is now available to be picked up by the user.
8. If the user types "go through $door$" then either move to the other room that the door connects to or tell the user that the door is locked.
9. If the user types "quit" then end the program. You can use "System.exit(0);" to quit.
10. If the user types "introduction" then print the introduction again.

Order of events:

1. Game starts, do the following:
    a. Print out your name
    b. Print out the introduction
    c. Print out the help message
2. Accept commands until the main door is open. When the main door is open the user wins and the game is over.

Additional requirements:

1. The main door to outside must be locked.
2. You must create a key that unlocks the main door.
3. You must create a breakable item that contains another item.
4. There has to be at least 4 items.

Rubric:

1	A "backpack" is created in the Game class. I suggest an ArrayList of type Item.
1	look command: shows the name and description of the current room, doors in the room, items in the room, and items in the user's backpack
1	look at thing command: prints out the name and description of the thing
1	pick up item command: an item is removed from room and put in the user's backpack
1	use item command: if is the key, then unlock the door (this leads to winning). This presumes you are in the main room. Otherwise, it does what you want it to.
1	throw item command: breaks the item if breakable, otherwise, takes the item out of the backpack and puts it in the current room
1	go through door command: changes the current room to the room on the other side of the door.
1	win: when the front door is unlocked the user wins
1	breakable item: an item can be broken when thrown
1	containable item: at least one item when broken contains another item