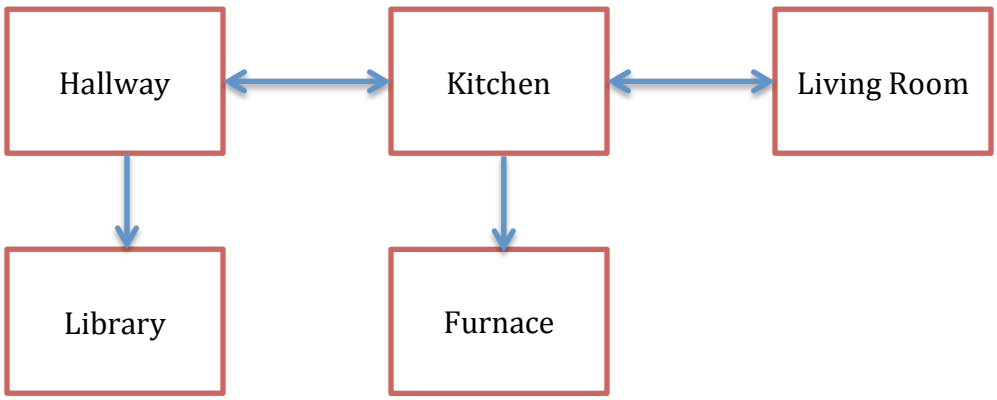
CSCI 1300 Introduction to Programming [Fall 2014]
Instructor: Boese
Assignment #3
Due: Friday **Sept 12, by 10am**.

A Dungeon Game!

Objectives:
There are a couple of objectives with this assignment.

1. Learn how to take a big problem and break it up into small steps to work on. For example, Step 1, Step 2, Step 3, Step 4, Step 5. No one, not even me, will start a big program (including writing up this assignment) by trying to make the whole thing work the first time. We break up the problem in small pieces and get that to work, then add to it, test, add to it, test, etc. until we complete the full problem.
2. Learn how to read an algorithm in pseudocode and translate it into Python.
3. Learn how to write loops.
4. Learn how to properly make use of if/elif statements.
5. Learn how to nest if statements.
6. Have fun.
7. An Extra Credit option that isn't easy to challenge those of you who are seeking more challenge in this course. No, the extra credit isn't worth many points. It is there to inspire you.

For this assignment, you will be creating a little dungeon game. You will design your game with the minimum of the following rooms in this format:



When you run the program, the user will type in the direction they want to go. For example, when you are in the kitchen, there are rooms to the west (w), south (s) and east (e). The user would then enter either the letter 'w' or 's' or 'e'. If the user enters 'w', then they move in to the hallway room. Now they have the option to go south or east. They continue playing the game until they either find the princess or they end up in the furnace. If they go to the furnace room, you will tell them that they died and the program ends. If they find the princess, *(who is in the library diligently studying the Python programming language)*, you will also end the program and tell the user that they are a hero! The last option, is that the user can type 'q' at any time and quit.

Here is an example run of the program. The user enters single characters only – in this example, the user typed in: e, s, s, e, n, s, w

```
Welcome to Liz's Dungeon!
You are in the kitchen. There are doors to the west (w), south (s) and east (e). w
You are in the hallway. There are doors to the south (s) and east (e). w
You are in the hallway. There are doors to the south (s) and east (e). n
You are in the hallway. There are doors to the south (s) and east (e). e
You are in the kitchen. There are doors to the west (w), south (s) and east (e). s
You entered the furnace and fry yourself to death!
Goodbye!
```

Here is another example run of the program:

```
Welcome to Liz's Dungeon!
You are in the kitchen. There are doors to the west (w), south (s) and east (e). w
You are in the hallway. There are doors to the south (s) and east (e). s
You are in the library. You found the Princess! You are a Hero!
Goodbye!
```

Requirements:

1. Name your file: **DungeonGame.py**
2. You must use `elif` where appropriate. This makes it more efficient than using separate if statements.
3. If at any time the user types in 'q', then the program should quit and say goodbye.

```
Welcome to Liz's Dungeon!
You are in the kitchen. There are doors to the west (w), south (s) and east (e). q
Goodbye!
```

4. You must implement the 5 rooms as shown in the diagram above. The arrows designate which way a room is available. For example, once you go to the furnace, there are no exits. You cannot change from this core room setup.
5. The output for the core rooms in the diagram must match the example output provided. The only exception is if you apply the extensions listed below.

If you meet the requirements specified above, you can get a 99 for the assignment. To get the final 1 point, you will need to add the following extension.

Extensions

1. [1 point, easy]: Extend the game with at least 3 additional rooms. You may only extend the game by going north from the Hallway. You can then do what ever you want from there, so long as the core room diagram above stays in tact. The princess must still be in the library, but if you want to have a twin princess, or another room of death somewhere, or… have fun with it in your extra rooms.

2. [Extra 2 point, hard]: Modify the library so that the user has to find the princess somewhere inside the room. You will need to loop while the user looks around the room until they find the princess. Follow the output as shown in the example below. The user should also still be able to type 'n' to leave the library and go to the hallway *(not depicted with arrows in the diagram because this is an extension to the core)*. The output and princess location must be exactly as demonstrated:

```
You are in the library. You sense the princess is here, but where? Look around (l) l
You look, and see a picture (p), bookcase (b), closet(c) x
You look, and see a picture (p), bookcase (b), closet(c) b
You look through the bookcase, and see your Python textbook. c
You rummage through the closet, and you find SKELETONS! p
You look at the picture, and see the princess hiding in it! You are a hero!
Goodbye!
```

How to get started

This is a bigger program than the previous ones, but what you will find is that once you get two or three rooms programmed, adding new rooms is relatively easy as it is similar code. So, the way to start programming this assignment is to begin with two rooms. For example, try to set up the kitchen and the furnace rooms.

Hint: to keep track of where the user is, add a `string` variable that specifies which room you are in. Then each time the user moves to a different room, change the variable to be the current room that they are in. So for example, if you start the program and you are in the kitchen, and the user types 'w' to go to the hallway, then change your variable that keeps track of the current room to be the hallway.

Step 1:

1. DISPLAY the welcome message to the user, where the name of the dungeon is your name.
2. SET a variable for the current room to be the kitchen, as you always start out in the kitchen.
3. SET a variable for keeping track of the command that the user will enter. Initialize this variable to an empty string.
4. LOOP while the command the user enters is not 'q'
    a. IF the current room is the kitchen
        i. PROMPT the user for which way to go: "You are in the kitchen. There are doors to the west (w), south(s) and east (e)."
        ii. READ in the command from the user
        iii. IF they enter 's' for south
            1. SET the current room to the furnace
    b. ELSE if the room is the furnace
        i. DISPLAY to the user that they have died: "You entered the furnace and fry yourself to death!"
        ii. SET the command to 'q' so the loop finishes
5. DISPLAY "Goodbye!" to the user

At this point, you should be able to run the game, though it is not a fun game with only 2 rooms! But the loop should work as in the following example run:

```
Welcome to Liz's Dungeon!
You are in the kitchen. There are doors to the west (w), south (s) and east (e). n
You are in the kitchen. There are doors to the west (w), south (s) and east (e). w
You are in the kitchen. There are doors to the west (w), south (s) and east (e). e
You are in the kitchen. There are doors to the west (w), south (s) and east (e). s
You entered the furnace and fry yourself to death!
Goodbye!
```

Step 2:
Add the hallway to the west of the kitchen. Test your program that you can go 'w' and see that you are in the hallway, then go 'e' to return back to the kitchen.

Step 3:
Add the library to the south of the hallway. Here the user will find the princess and win the game! Be sure the program quits once you find the princess.

Step 4:
Add the living room.

Step 5:
Add the extensions, if desired.

Grading:

Points	Description
60	**Interview Grade: Explaining your program**
15	Step 1
5	Step 2
5	Step 3
5	Step 4
5	Use of elif appropriately
4	The output is exactly as specified
1	Add your own additional rooms north of the kitchen
2	Find the princess hidden in the library
102	**Total points available**