Programming Assignment 5

Start Assignment

Due Thursday by 11:59pm **Points** 100 **Submitting** a file upload **File Types** zip

DUNGEON ADVENTURE

PROGRAM PURPOSE:

The purpose of this is assignment is for you to practice working with composition (has-a relationship), inheritance (is-a relationship), modular design, and two dimensional arrays as well as to continue developing your skills at working with multiple classes.

For this assignment, you will utilize your classes from Heroes versus Monsters and modify them as necessary to fit the requirements of this assignment. You will also write three new classes: Room, Dungeon, and DungeonAdventure. You are welcome to provide additional classes in your solution as you see fit.

PROGRAM DETAIL:

This is an adventure game where a hero is randomly placed within a dungeon that is (at least) a 5 x 5 2D array of Rooms. The hero needs to find the two pieces of the Crown of Coding, and take them to the exit to win the game. Some features of the dungeon will prove a hindrance to the hero's task (monsters!), while some will prove helpful (healing potions). Your task is to write a correct and well documented java program that will simulate this adventure. NOTE: You are welcome to implement a variation on this theme provided you adhere to the spirit of what is being asked on this assignment.

CLASS DETAILS:

Room.java

- Contains default constructor and all methods you deem necessary -- modular design is CRUCIAL
- Contains the following items/behaviors
  - (Possibly a) Healing Potion - heals 5-15 hit points (this amount will be randomly generated -- you can modify the range)
  - (Possibly a) Monster – If a room has a monster, randomly generate which one
  - EXTRA CREDIT IF WORKING ALONE (Possibly a) Pit - damage a pit can cause is from 1-20 hit points (this amount will be randomly generated - you can modify the range)
  - EXTRA CREDIT IF WORKING ALONE (Possibly a) Vision Potion - can be used to allow user to see eight rooms surrounding current room as well as current room (location in maze may cause less than 8 to be displayed)

- (Possibly an) Entrance - only one room will have an entrance and the room that contains the entrance will contain NOTHING else
- (Possibly an) Exit - only one room will have an exit and the room that contains the exit will contain NOTHING else
- (Possibly a) Crown Piece - two pieces in game and they will never be in the same room
- Doors - N, S, E, W (depending on room location some directions can be disabled if you wish)
- Room content possibilities (you may modify these values to make your game more realistic/challenging
  - 10% possibility room will contain a healing potion
  - 5% possibility room will contain a vision potion (remember this is extra credit if you are working alone)
  - 10% possibility room will contain a pit (remember this is extra credit if you are working alone)
  - 10% possibility room will contain a monster

- Must contain a toString method that builds a 2D Graphical representation of the room (NOTE: you may use any graphical components in Java that you wish). A possible command line representation is as follows:
  - * - * will represent a north/south door (the - represents the door). If the room is on a boundary of the maze (upper or lower), then that will be represented with ***
  - East/west doors will be represented in a similar fashion with the door being the | character as opposed to a -.
  - In the center of the room you will display a letter that represents what the room contains. Here are the letters to use and what they represent:
    - X - Monster
    - P - Pit
    - I - Entrance (In)
    - O - Exit (Out)
    - V - Vision Potion
    - H - Healing Potion
    - E - Empty Room
    - M – Multiple Items

Example: Room 1,1 might look like

```
* - *
| P  |
* - *
```

Room 0,0 might look like

```
* * *
* E |
* - *
```

Hero.java (update as necessary)

- Contains an explicit value constructor for the name of the hero
- Contains at least the following:
  - Hit Points - initially set to 75 - 100 upon creation (randomly generate - you can change the range)
  - The number of Healing Potions
  - The number of Vision Potions
  - The number of Crown Pieces found
- Increases or decreases the Hit Points accordingly
- Contains a toString method that builds a String containing:
  - Name
  - HitPoints
  - Total Healing Potions
  - Total Vision Potions
  - Total Crown Pieces Found

NOTE: The Hero and the Dungeon will need to interact. When the Hero walks into a room if there is a potion in the room, the Hero automatically picks up the potion. Likewise if there is a pit in the room, the Hero automatically falls in the pit and takes a Hit Point loss. All of these changes obviously affect the room. For example, the Hero walks into a room that contains a Healing Potion. The Hero will pick up the potion, changing the Hero's potion total, as well as changing the room's potion total.

Dungeon.java

- Creates/contains a 5 X 5 2D Array of Rooms (you can make this larger if you wish)
- At construction time places the Entrance, the Exit, and the Crown Pieces. NOTES: the entrance and exit are empty rooms. The crown pieces cannot be at the entrance or the exit and cannot be in the same room as one another.
- (possibly) Maintains location of the Hero in the Dungeon
- Contains a toString method that builds a String containing information about the entire dungeon (all the rooms as well as location of hero).
- Anything else you deem necessary

DungeonAdventure.java

- Contains the main method
- Provides an introduction to the game describing what the game is about and how to play
- Creates a Dungeon Object and a Hero Object (based on selection by player)
- Does the following repetitively:
  - Prints the current room (this is based on the Hero's current location)
  - Determines the Hero's options (Move, Use a Potion)
  - Continues this process until the Hero wins or dies
  - NOTE: Include a hidden menu option for testing that prints out the entire Dungeon -- specify what the menu option is in your documentation for the DungeonAdventure class
- At the conclusion of the game, display the entire Dungeon

- NOTES:
  - You do not have to support the ‘Run Away’ feature of Heroes versus Monsters
  - You do not have to provide a feature that allows the player to play again
  - The other mechanics of battle from Heroes versus Monsters (attacks per round, special skills, etc.) should be supported

WORKING AS A TEAM:

You may work in teams of up to three students. If you work in a team, you will turn in a **DETAILED** account of each team member's contribution in your readme.txt. There is enough work to go around where each team member should be responsible for a class, but it is fine to work together on everything if you wish. Separating concerns is great practice for future classes and industry.

An important consideration when working as a team is how to share code. Utilizing git/GitHub can be very helpful if you have the time to get up to speed. Tom has recorded lectures and is glad to meet with you to go over git if you would like. You can also use things like Google Docs, DropBox, or One Drive as ways to share.

EXTRA CREDIT (for single person):

(10 points) Implement an additional potion - the Vision Potion. The Vision Potion allows you to see the rooms that are immediately around you (this is up to eight rooms depending on your location in the dungeon). This potion only lasts for a single turn. Example:

The hero is currently in room 1,1. If the hero drinks the Vision Potion, then the following rooms are visible for a single turn.

Room 0,0 Room 0,1 Room 0,2
Room 1,0 Room 1,1 Room 1,2
Room 2,0 Room 2,1 Room 2,2

(5 points) Add pits as a possibility for each room. Note they are like the opposite of a healing potion and easier to deal with since you don’t have to keep count of how many the hero has 😊

EXTRA CREDIT (for teams):

(Up to 15 points) MAY be given for additional features (playing sounds, loading images, originality, creativity, etc.). This is at the discretion of the grader so is not guaranteed. If you feel you did things that warrant extra credit, clearly describe these things as part of your readme.txt.

Of course, the wise students will first get the required features of the program implemented recognizing that the assignment is quite involved!

TO TURN IN:

- A zip file named with your Net ID that contains

- All .java files
  - Make sure Javadoc comments are used
  - Make sure you follow coding style (final params, private fields, etc.) and naming conventions (my for fields, the for params, etc.)
- A readme.txt that provides
  - name(s),
  - estimate of time to complete assignment
  - for teams, discuss what each person worked on
  - discussion of extra credit
  - list of shortcomings
  - any questions you have

Dungeon Adventure Rubric

Criteria	Ratings	Pts
Misc -proper files -readme.txt that describes work done, time spent, shortcomings, etc.		10 pts
Dungeon Class -holds rooms, generates dungeon, manages items in dungeon		20 pts
Room class holds necessary items, allows for updating as necessary		10 pts
Hero class updated to count potions, crown pieces, etc.		10 pts
DungeonAdventure -allows for gameplay as specified, interacts with user, displays necessary info in clear fashion		20 pts
Naming conventions, visibility, coding style, documentation -fields named with my -params named with the -params final -Javadoc thorough and clear		15 pts
Game play game plays as specified, is understandable, can win/lose		15 pts
		Total Points: 100