Project 1 CSc 421 Kutztown

Purpose: Java Review. Language elements, including objects. Design patterns.

Due: TBD, in D2L and via the turnin script. Late, emailed, or D2L code submissions will not be accepted.
Setup and test *turnin*. Inability to use *turnin* is not an excuse for late submission.

Description: Games with objects have been played in many different forms. For this term's projects, you will implement a game of your choice where one or more players compete.

You will write a complete program that represents a game for one or more players. The game must use objects, e.g. cards, dice, pieces, etc.; This first project will be a text-based application, run using the java interpreter on acad's command line (or a PC's).

You will define object types for this program. One must be for a singular playing object, e.g. a die, a card, a game piece. Others will be determined by your design of the program.

Each member of CSc 421 will implement a different game. Research games played with objects; many games can be found online that you can play if you aren't familiar with them, although it is likely better to implement a game with which you are familiar.

Once you have decided upon a game, you must claim it on the class forum. Choices are strictly first come first serve; as soon as you have an idea, post it in the forum thread designated for term project choices. Within this post, provide the following:

- Name
- Minimum and maximum number of players
    - Maximum of two players might be best so things don't get out of hand
- Object(s) with which it is played
- Link to online version
- Link to rules
    - Note any deviations you intend to employ

If you intend to implement in any other language but Java, include that info and the tool(s) you will use to implement it as a web application.

Once you have settled on a game, get to work on a design, which we will discuss this further in class. Design patterns can be useful in this thought process. For example, there will be a main class that controls the objects of your program. Consulting literature about the **controller** design pattern (among others) is recommended. Here's a good source:

https://www.tutorialspoint.com/design_pattern/design_pattern_overview.htm

Remember, you must claim your game in the forum; be sure nobody has claimed it before you.

Submit your completed design in the Project 1 Design drop box on D2L and wait for my feedback before you begin coding the project.

Notes:

- You will be informed if there are any issues with your choice of game asap after you post it.
- Designs MUST be approved before coding can commence.
    - Alternatives to a Java application can be considered, but only with permission. Note that we will be using a Java framework in subsequent projects (although alternatives may be permissible).
- The overall game layout may be global in the main class, as it is likely that this class will later morph into a GUI.
- You must use printf to line up numbers properly.
- You must employ a method from classes in the instructor's Input package.
- Show game progress by some reliable means.
- All object-type classes MUST have appropriate set and get methods. This may save you some work later.
- Keep in mind that this project will be repeated several times this term, with all subsequent versions window based. You should be cognizant of this fact while making design decisions for this project.
- Turn in a file, named **readme.txt**, (all lower case, this exact name) that augments your design, along with other information you would provide to someone who purchased your product (including how to play)
- This is a senior/graduate level course. Proper style and modular design is a must. Substantial penalties, up to and including your program not being graded, will be levied for lazy, incomplete, or chintzy style, and/or not employing a proper modular design.
- **Graduate Students Only**: Log the results of each game play in a file named **log.txt**. When the user starts a game, print historic information, which can include how many games have been played, and the maximum, minimum, and average stats (number of turns required?), as stored within this history, when the game starts.

Deliverables: Deadlines TBD during class.
Design: System design artifact(s) in Project 1 Design D2L dropbox.
Project:

- All .java files via turnin script. Do not turnin the Input pkg (but you must properly access at least one input routine within the package).
- Submit your readme, named **readme** with file type .txt or .pdf in the *Project 1 Design* D2L dropbox.
    - Include the names of all design patterns employed with a brief description of its application.