Tony Burwinkel
CS5004 Fall 2022
Final Project Concept Mapping

Concept Level(Basic, Intermediate, Advanced)	Concept	Where to find it	How it was demonstrated
Intermediate	Abstract Classes and Interfaces	Deck DungeonRoom Room Boss	Deck is an abstract class that accepts a generic type. Room and Boss classes are implementations of the DungeonClass interface. This interface makes polymorphism possible for traversal of the Dungeon by Heroes.
Basic	Recursion	DungeonRoom Room Boss	Room defendAgainst method recurses to the nextRoom until bottoming out at the base case, the Boss. This method mutates the dungeon by causing damage or giving coins to the boss, and it also returns a string of each hero attacking the dungeon so we know what happened. Room show method adds formatted text separated by arrows to show the dungeon. Bottoms out at Boss, which adds its representation and returns the string up the chain
Basic	Linked Lists	Dungeon class	Dungeons are linked lists of rooms that implement DungeonRoom. Similar to Node/EmptyNode. Rooms have a next field, which must be a DungeonRoom. Boss is like an empty node in that it finishes recursive method calls and returns them up the chain.
Intermediate	Equality and Comparison	Room class	Rooms implement Comparable so that the player's hand can be sorted by Sign type. The compareTo method delegates room comparison to comparing the Strings that are the room's sign.name().
Basic	Exception Handling	Controller getChoice Hero setHP	Controller uses try catch to handle IOExceptions. getChoice controller method catches input mismatches to ensure only integers are

			submitted by the user setHP method for heroes ensures negative health settings throw illegalArgumentException
Basic	Generics	Deck class	All decks can accept one kind of entity (Hero, Room, Boss). The deck class is abstract, defining common deck methods like shuffle and draw, while allowing different kinds of objects to occupy the deck.
Intermediate	MVC Design	ModelBM ControllerBM	ControllerBM accepts a Readable and Appendable. ModelBM class implements all game logic. ControllerBM's go method uses model's logic to play the game, and outputs results to its view (System.out or other appendable)
Intermediate	Composite Design	Player class Model class	Player is composed of several collections as well as a dungeon. Model is composed of two Player objects, a Tavern object, and 3 Deck objects.
Intermediate	Iterator Design	Dungeon, DungeonRoom Room, Boss	Dungeon implements iterable. Room and Boss implement the canAdvance() and advance() methods required, similar to the functioning of a Node and EmptyNode. This iterable/iterator implementation has been used to simplify calculating Sign totals for the dungeon, which determines which heroes will go where. The method call that uses iteration is in the Dungeon class (getNumSigns)
Intermediate	Adapter Design	Deck class	The Deck class wraps a stack implementation and delegates some functionality to it. The deck class chooses which methods to expose, while hiding some others from the user. For instance, we can pop from the stack of cards by using draw, but we can't select from the middle of the deck, which would be illegal for a deck but possible for a stack implementation. Similar to the idea of a Pull MVC, where the view gets a model adapter that does not allow mutation of the model.