1. What data will the system deal with?

- There will be an interface for all of the pieces, and a class for each of them independently that will implement the “piece” interface
    - This means that there will be a class called “King,” “Queen,” “Knight,” etc.
- There will be an Enum representing black/white within the interface.
- There will be another Enum representing the type of movement that a piece can do
    - Example: diagonal, straight, etc.
- Some of the instance variables that will be used within each class will include a reference to its location
    - The locations that it can move to will be stored in a set, and when a move is attempted it will be checked if that square is stored in this set.
- The board itself will be composed of an 8x8 2d array.

Classes that implement the piece interface:

- A public method will be made available within each class to move the piece
- A private method will also be used to display possible positions that the piece can be moved to on the board, based on the “movement” enum.
    - The way this will work is based on the “movement” of the piece at hand, a private method will calculate all squares within the array leading from the piece’s current location up to and including the first instance of another piece obstructing further movement, or until it hits the edge of the board; all such locations will be considered valid moves.
- A private method will be used to remove a piece from the board if its space is taken by another (opposing) piece, as this means the original piece was captured.
    - This method will be called when a piece is moved to another location, as it will need to check if a previous piece was here, and if there was a previous piece then it needs to remove it.