Player is parent superclass

Pitcher and Batter are child subclasses

3 classes total

any attributes/ code that is common is placed in the base class to share

Feel free to add helper methods to reduce code duplication.

BASE CLASS Player:

Add a private attribute called \_\_name ( name of the player)  
Add a private attribute called \_\_position ( position of the player on team)

Constructor

takes **two arguments** - name and position

It sets the corresponding private attributes based on these arguments

It should validate and throw a ValueError exception if either is empty string

Add two read-only public properties (called name and position) backed by the  
corresponding private attributes.

Add a getStats method

takes no arguments

returns a string of the form “Player Name: name Position: position ”.

CHILD CLASS Pitcher: Add Pitcher class derived from the Player

Add two public attributes - wins and loss – (number of game wins and losses credited to the pitcher)

Add a constructor

takes **three parameters**: name, wins and loss

The constructor should set corresponding attributes

It should throw an exception if the name is empty

or wins is < 0.

or loss arguments is < 0.

It should call the base class constructor to set the name and the position attribute (as “Pitcher”) in the base class.

Add a getStats method

returns a string that says: “Player Name: name Position: Pitcher Stats: wins-loss win-loss.”

the value of name, wins and loss come from its corresponding attributes.

This method should **use the getStats method of the base** class method.

CHILD CLASS Batter: add Batter class derived from Player

Add two public attributes:

at\_bats (number of times batter got a turn batting against a pitcher)

hits (number of hits for the batter)

Add a constructor

takes **four parameters**: name, position, at\_bats and hits

The constructor should throw an exception if:

either name or position is an empty string

either at\_bats or hits is negative

Add a read-only public property called average

returns the batter’s batting average: average = hits / at\_bats

Add a getStats method

returns a string that says: “Player Name: name Position: position Stats: Batting Avg: avg.”

The value of name, position, and avg come from the corresponding properties.

This method should **use the getStats method of the base** (PLAYER) class method.