Chapter 14

End of Chapter Exercises

1. *Add a method to the `Player` class (Figure 14.1) called `WinLoseRatio` which has no parameters but which returns a real value which is ratio of the number of wins to losses (i.e. `numberGamesWon ÷ numberGamesLost`).*

    ```
    METHOD WinLoseRatio RETURNS real
        IF (numberGamesLost > 0) AND (numberGamesWon > 0)
            RETURN numberGamesWon ÷ numberGamesLost ;
        ELSE
            RETURN numberGamesWon ;
        ENDIF
    ENDMETHOD
    ```

2. *Assume the `Player` object `rxforwar` has already been instantiated and the real player has played a number of online games. Write an `IF…ELSE` statement which calls `rxforwar`'s `WinLoseRatio` method (see exercise 1 above) and displays "Congratulations, you have a good Win:Lose ratio" if the ratio is greater than 1.0, otherwise "Sorry, your Win:Lose ratio is low and you need more practice".*

    ```
    IF rxforwar.WinLoseRatio > 1.0
        Display ('Congratulations, you have a good Win:Lose ratio') ;
    ELSE
        Display (' Sorry, your Win:Lose ratio is low and you need
    more practice') ;
    ENDIF
    ```

3. *Say we learn that the game has been modified so that there are now two classes of player: normal and league. The Win:Lose ratio for normal players is the same as before, but for league players (those who have signed up for a premium content game with competitions and prizes) the Win:Lose ratio is only based on the number of league games lost and won. League players may join 'normal' players in a game but such games will not count towards their ratio. Declare a new class `LeaguePlayer` which inherits from `Player` but overrides the `WinLoseRatio` class to conform to the new system.*

    ```
    CLASS:LeaguePlayer EXTENDS Player
        // Properties
        string:name ;
        integer:numLeageGamesPlayed,
                numLeagueGamesWon,
                numLeagueGamesLost ;


        // Constructors
    ```

```
METHOD Player (string:aName)
   Super (aName) ;
ENDMETHOD
. . .
METHOD WinLoseRatio RETURNS real
   IF (numLeagueGamesLost > 0) AND (numLeagueGamesWon > 0)
      RETURN numLeagueGamesWon ÷ numLeagueGamesLost ;
   ELSE
      RETURN numLeagueGamesWon ;
   ENDIF
ENDMETHOD


METHOD WinLoseRatio RETURNS real
// rest of the methods, as before
. . .
ENDCLASS
```

Projects

No solutions provided as they will be highly dependent on how you have structured your own solutions over the previous chapters.