# Key of what each color means

Unchanged / nothing new.

A ToDo

Movement of writeup, changed wording, new.

Comment.

I have Added additional outlining to have more levels meaning easier navigation and keeps me to the title to prevent waffle and forcing me to write about the correct thing.

Write up that has been deleted will not be marked.

# Analysis

## Aim:

The aim of my computing project is to create an Golf application which can record scores from matches following various golf match types, i.e Stableford, matchplay and strokeplay. The golf application should be for casual players who just want to have fun, record their matches and see what their golfing friends are up to.

## Identification of project:

Currently at school match level the teams record the scores on score cards which are pieces of cards. These can be a problem because they are easily lost, damaged and can’t cope with rain. Personally, I believe that an online version will be much better as a lot more data can be stored. This is the same belief as Mr. H, who is a professional golf coach at Stowe. He will be acting as a client for this project. In addition, as a casual player it’s very hard to see where certain people might be playing. This problem shall also be incorporated into the new application so that you can have “Friends” and if they are currently in a match you can see which club they are playing at, the current score in that match and who they are playing against, if they are part of the new application. Another aspect of the old card-based system is the score calculation. A lot of the golf match types can be confusing and involve a lot of confusing math which can lead to mistakes, with an online computer-based system this means that the computer will accurately calculate the score for each hole without failure, therefore speeding up the game allowing more people to play per day. Finally, a way in which the new system shall help golfers is easy access to the clubs with a hyperlink to their websites, you can see how many holes there are, how long each hole is, a rating into how good other golfers think the golf course is and a short description on the golf course. These are all possible data to add as long with tons of other information.

## What’s currently out there?

Within my research I have gone on to the app store, downloaded and tested multiple apps to see what ideas they have and what makes a strong application as a user perspective. Personally, I am not a golfer meaning I have asked for help in judging these applications from Golfers.

### Golf application 1: “Distance Calculator” creator, “Qi Chen”.

Although this isn’t the golfing application as previously mentioned it proposes a very interesting design idea. It works by taking a photo of the golf flag you then crop the photo so that it’s the height of the bottom of the flag to the top and uses the size of the flag to calculate the distance till the hole. This application requires a calibration to set up the correct distance and I’m unsure of the accuracy of application, but it is still a very interesting idea and one to take into consideration when thinking of what to add to really make the product stand out.

### Golf application 2: “Golf Game Book” creator, “GameBook Oy”.

This is more like the design of project as I originally started with and contains much of what I initially thought up of. It contains a strong social media section where you can post photos videos, golf tips and much more. This is much more advanced, but it might be interesting to add a few 5-minute golfing videos for people to improve their game. Personally, I feel that the capability to add photos and videos personally might be unnecessary for my application, but I can speak with my client and see what he thinks. This application also includes an ability to see which friends are on a course practicing or playing a match/tournament this a very beneficial concept to the application and is one of the most valuable credits. The scoring section of the game is very in depth with thousands of clubs and courses on its database each with various amounts of data. Upon selecting a course, you then have a choice of which type of match play you are playing out of the extensive list you then add your players including members and temporary players who don’t have an account. Finally, you add the scores for each player at each hole and when you finish it shows the winner. The only problem is the lack of data in match and on the club. This is something that I shall aim to improve when creating my version of a similar application.

### Golfer application 3: “Golf Weather” creator, “GolfWeather”.

Does exactly what the name suggest. It tells you the weather at that certain course for a week. It only tells you the weather for 6:00, 9:00, 12:00, 15:00, 18:00, the information is the condition at that time the temperature wind speed and whether there is going to be rain. It also tells you Sunrise and Sunset. These are all very important sets of information that are important in deciding whether to play a game. This could be an interesting to incorporate into the project although I’m not sure how I would be able to keep it up to date without doing it manually. There is a also a function called course near me which takes your location and finds all the courses near that location.

### Golfer application 4: “Golf Breaks Sunshine” creator, “Sunshine Golf Breaks Ltd”

This is another interesting idea that could be incorporated. It is a travel app which has a selection of pre-packaged holidays which vary in duration, and amount of people, which are all based at Golf club and containing rounds of golfs. It also has a selection of Green Fees at famous clubs with buggies or walking. This aspect could be added to the object: Course.

### Golfer application 5: “Golf GPS Range Finder Simple” creator, “Bryan Thornbury”

This range finder works very different to the first one but is still interesting. It works by presenting a google earth view. From this view you can see the golf course and your current position. The next step is to press a position on the map and it will mark the line from you to where you press and marks the length in yards you can then press another position and it will calculate the distance from the new position marked and the previous one. This can be done as many times as wanted and in the top right it has the total distance. This is perfect for recording the length of multiple shots. When your done you can press the rubber and it would delete all the lines you have made.

For my project it could be quite cool to add some video of Mr H or Mr G doing some coaching.

## User Needs

• To create a profile as a golfer.

• To be able to add more data to the system.

• To be able to create a match and send invites to offer golfers to join the match.

• To be able to add a score for each hole in the match.

• The system must be able to work out the winner of the tournament.

• To be able to add a “friend”.

• To be able to send a “friend” a message.

## Possible Additions

# Design

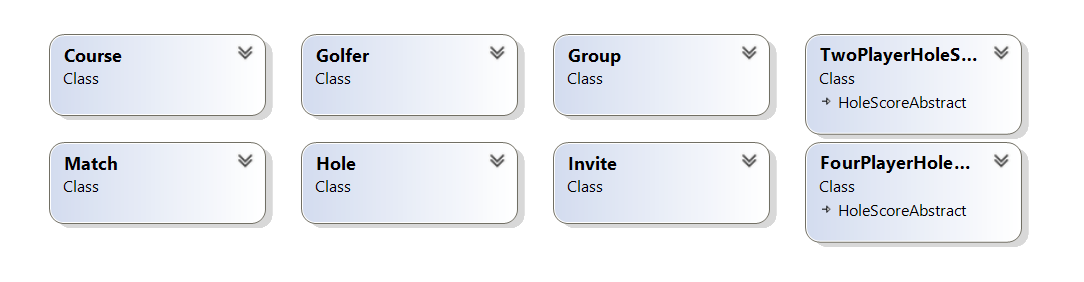
## Why I chose an Object-Oriented approach?

Object Oriented coding is based purely on the concept of “Objects”. This means that you can store the data that the physical object would have, just like the sheet of paper that you would write you’re scores on. In the program these objects are fields. The most important reason that I chose to use object-oriented Programming techniques is the ability to call methods on an object. For example, “Create A New Golfer” which creates a new instance of the class / object of golfer. This means that it is just like the physical object and makes it very useful in thinking what the main functionality requires. You can also extract a lot of data on objects as it is all in the same place. In addition, as the client if they don’t understand programming it can be easier for them to comprehend what is happening.

What is an object made up of, how do you define an object, what do objects allow you to do.

## Why I chose to Naked Objects.

## Class Diagram showing my objects



This diagram is new.

Objects:

Course: This is just as the name suggests it’s a course. Each course has a few properties that are just data about that course for example a name location address and so on. Course also has a collection of holes this means there is a one to many relationships between course and holes. I have made Hole a separate object because that its self has data about it and its own functions.

Golfer: A golfer is the account that you as a user will create when joining the application. You will need to add handicap but will not have to add properties like emails are optional.

Hole: As mentioned before holes interact with course but other than that they have no method on them self just optional data. This will effective be the exact some rows that are on a paper score card.

HoleScores: this score for each hole and portrayed as a table within the interface. This is the actual score sheet that you can see. There is two objects since match play only has 2 players but stableford and strokeplay has 4 players.

Match: Match is the super class for the other match types it contains the collection of golfers showing another many to one relationship as a golfer can only be in one match at a time. It also ahas the one to one relationship for course.

Group: is a collection of golfers with a group owner and a collection of members this just allows quick access to your friends or people you play with on a regular basis, there can also be a collection of messages that everyone in the group can see.

Invite: This is just a Boolean that says whether to accept to be in that match.

## Class Diagram showing inheritance for matches.

### What Inheritance allows me to do

### Inheritance inside my program.

This class diagram shows that I have deicded that MatchPlay,MatchStableFord and MatchStrokePlay all inherit data and behavior from parent class. This parent class is “Match” and is also known as the super class. The written terms this effectively means MatchPlay “is a “Match and the same for MatchStableFord and MatchStrokePlay. This is helpful in when they all have a certain same method and there for I can reduce the code duplication.

Add Golfer inheritance

For each of the main objects there is also a Configuration class which creates new instances and or methods like show all golfers.

## Class Diagram showing associations.

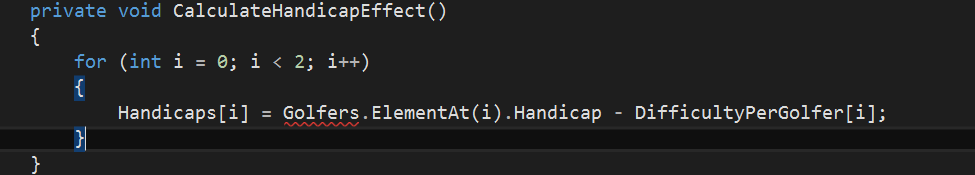
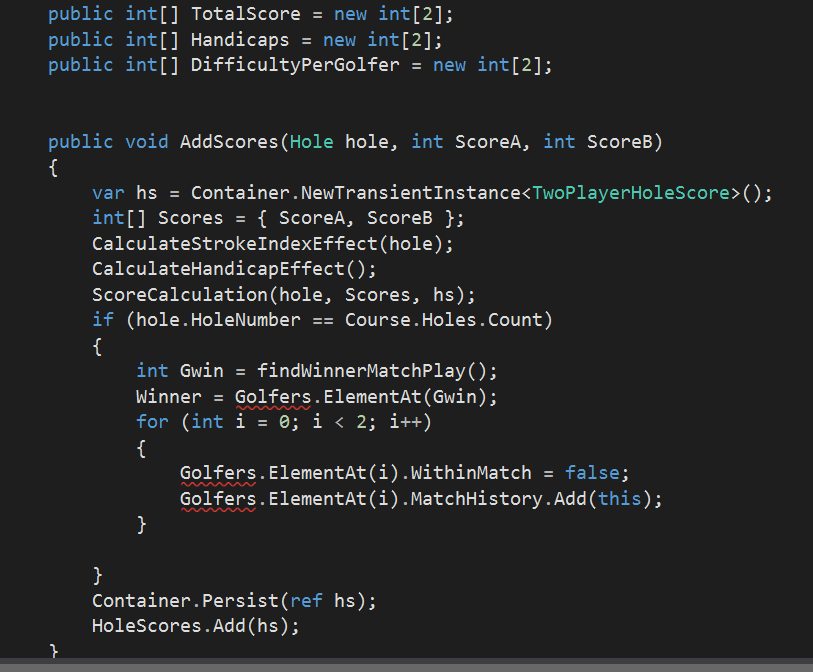
### What is an association?

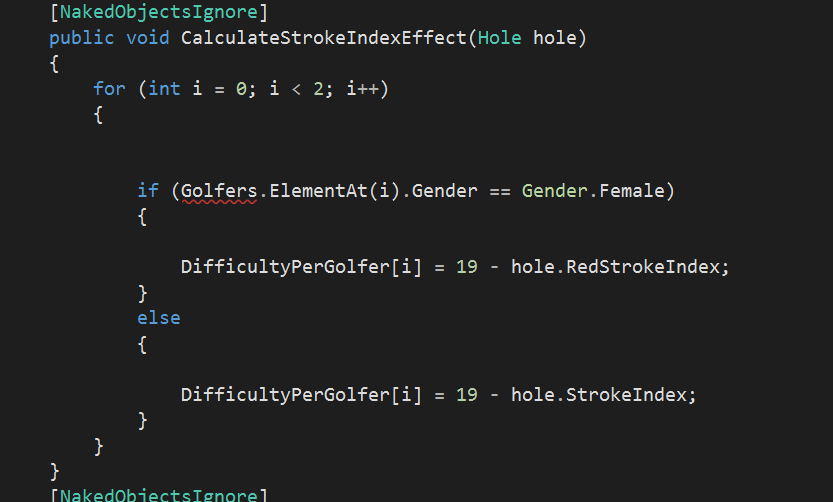
## Structure of Database.

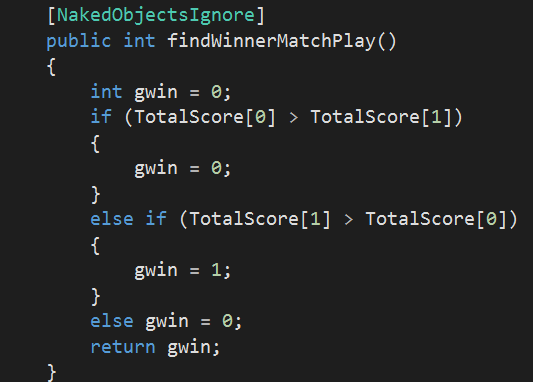
# Technical Solution

## Algorithms

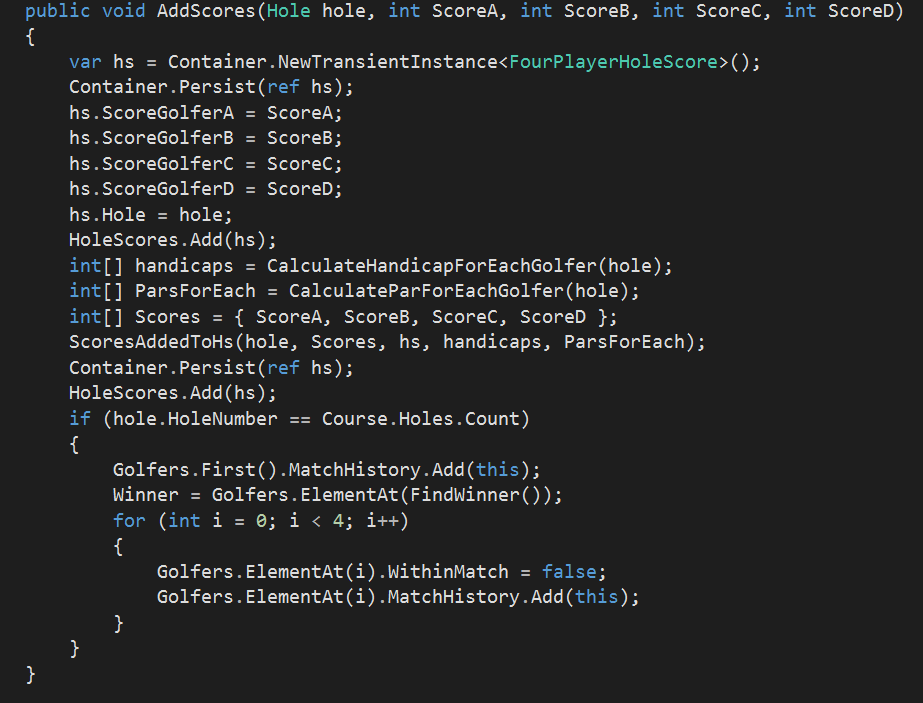
### MatchPlay

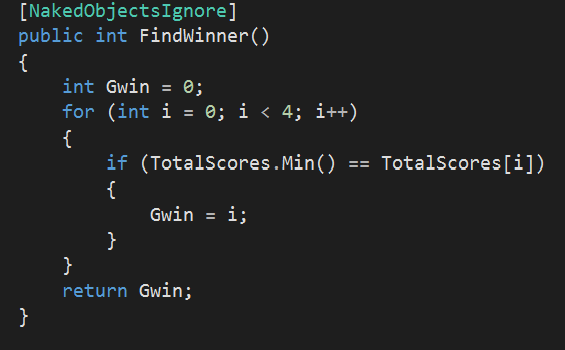


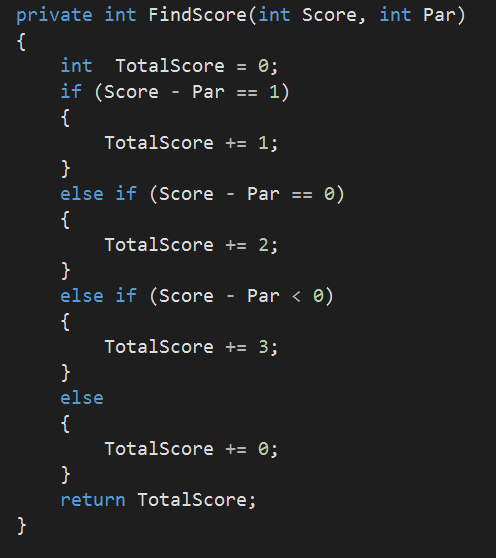
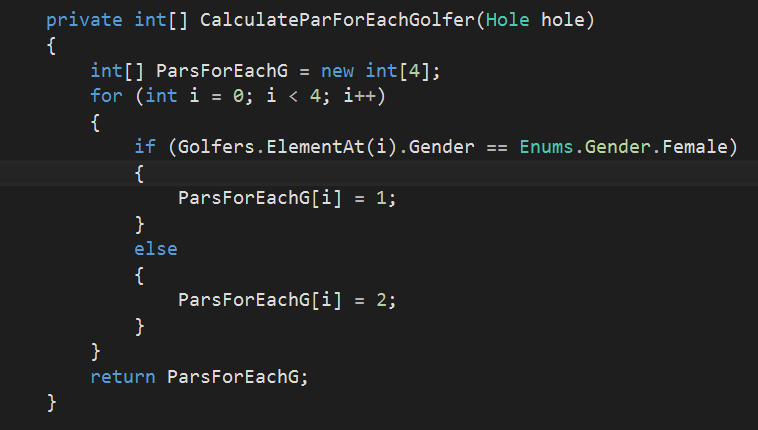


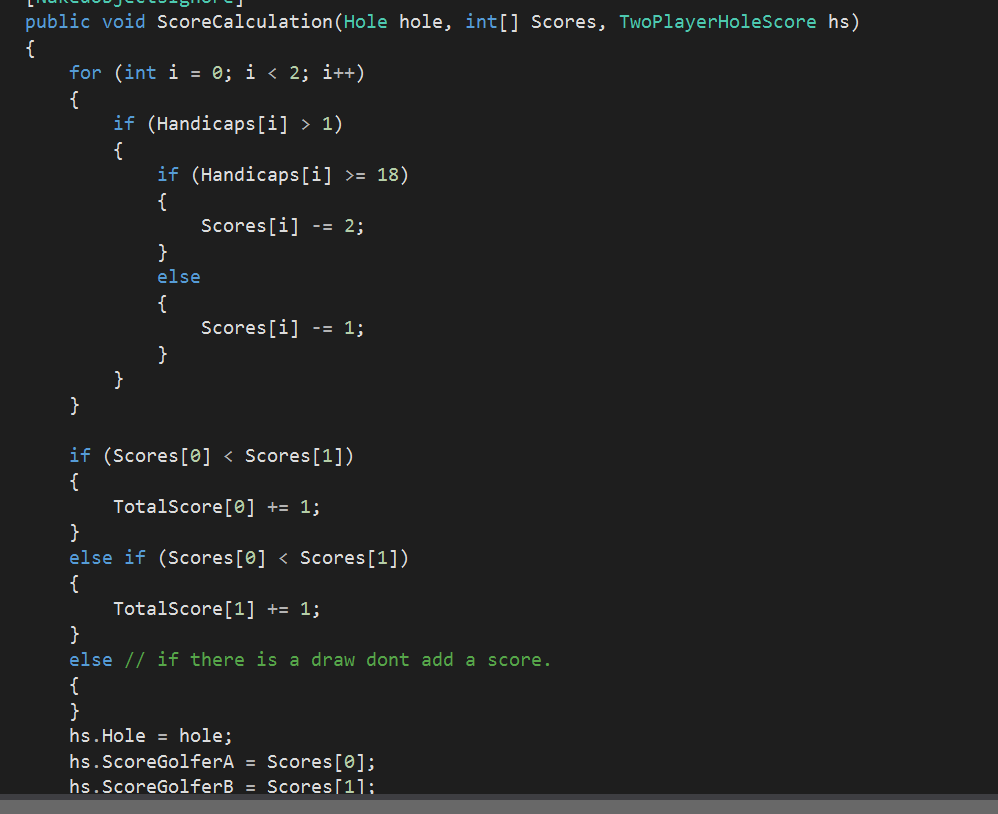
Annotate Code

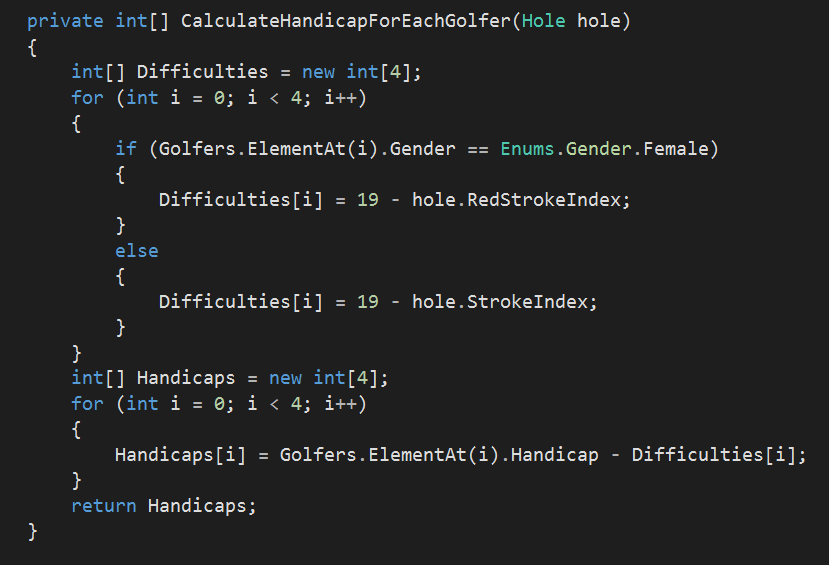
### Stableford

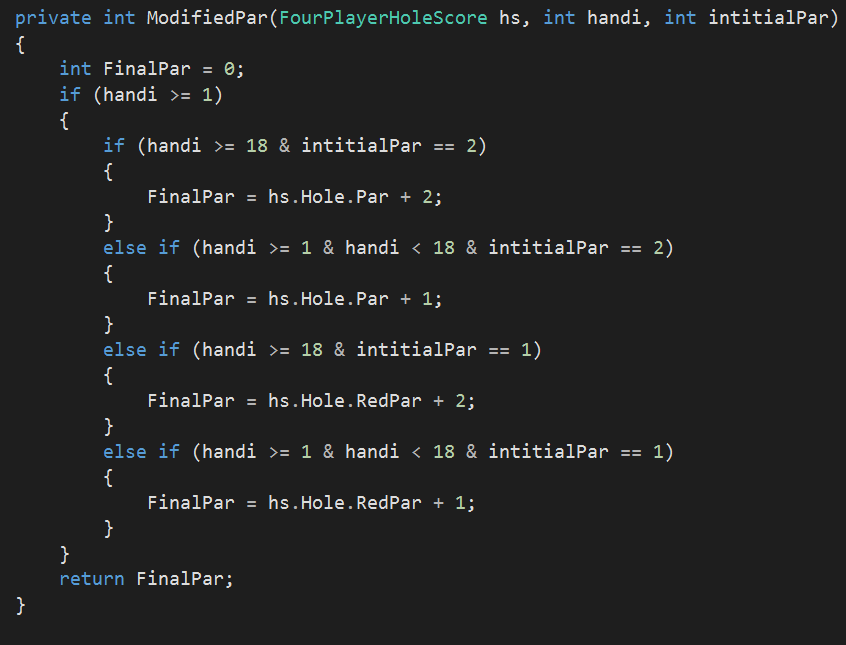


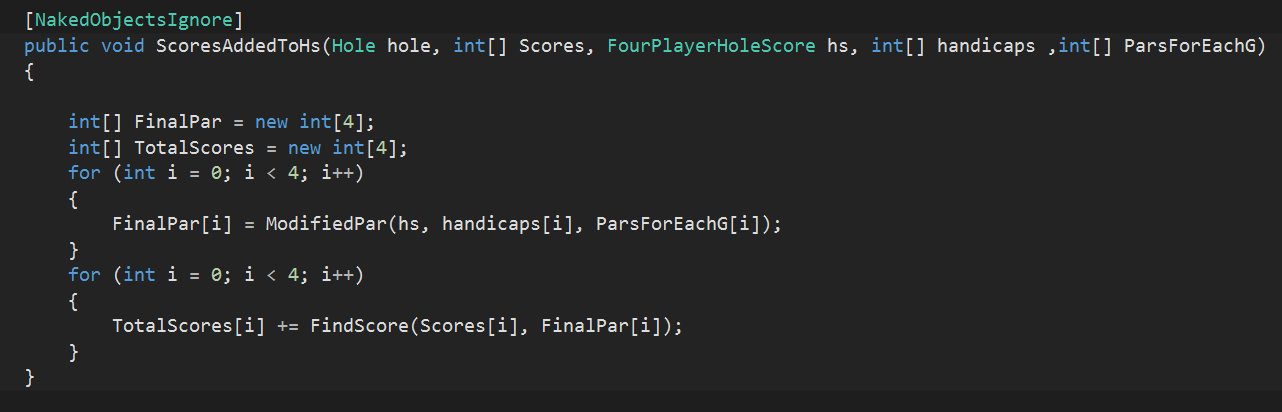


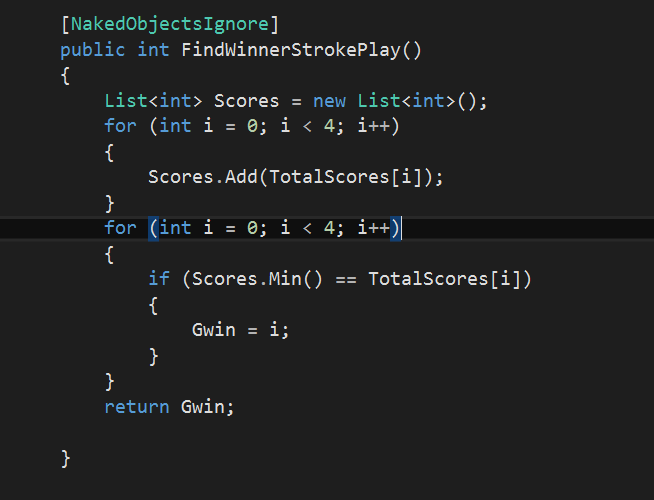
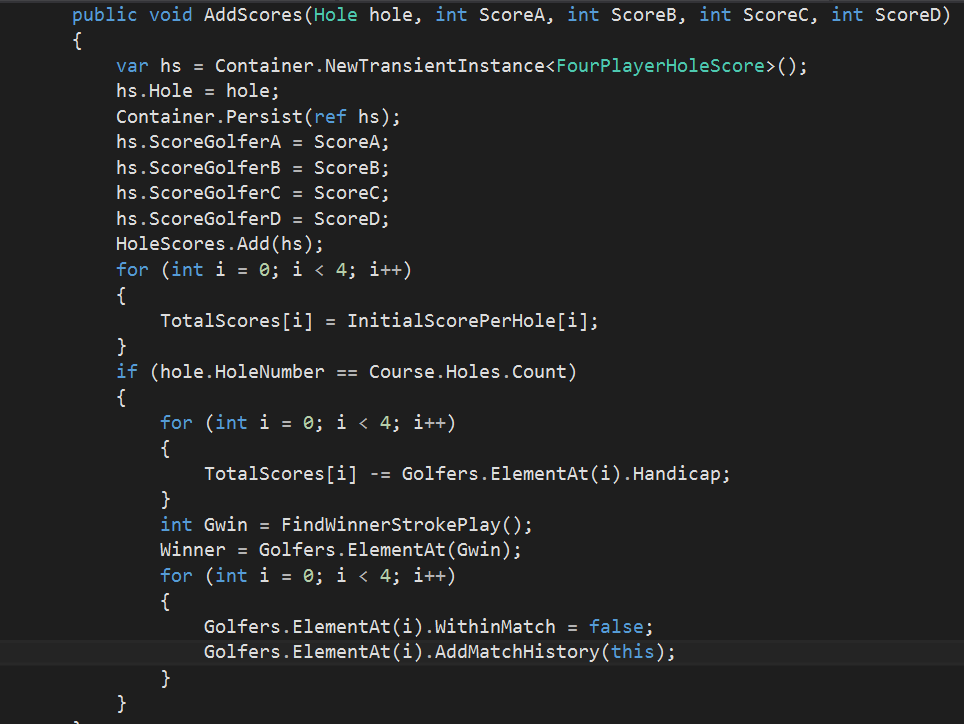








### Strokeplay

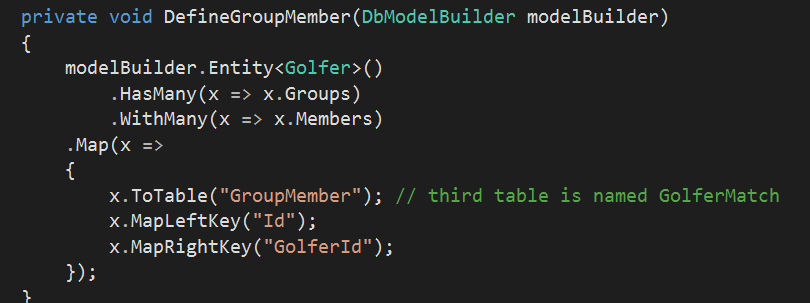


## DataBase Mapping

### Manual Many To Many

### 

### Manual many to self.



Must Describe all the code and how it works, why I did what I did.

# Testing

# Evaluation