

**Group Members:** Sofia McKerrow, Calvin Schnierer

**Student IDs:** s3626804, s3627341

# **COSC1295 Advanced Programming Assignment 1**

## **Github Repository**

<https://github.com/rmit-s3627341-calvin-schnierer/advanced-programming-ass1-2017>

## **Design Questions**

1. The Database class reads in a list of Athletes and Officials from the athletes.csv and officials.csv files. It also has an ArrayList of Games which serves as the record of each game. A game stores the list of Athletes in a race, a race Official and an Athlete that is the user's predicated winner.
2. Participant is an abstract class so a user cannot create a Participant object. Furthermore, a generic Athlete object cannot compete in a race as an Athlete object would return a false Boolean value from the canRaceInGame() method.
3. Database has an ArrayList of Games. From the Driver class the user can create a game which will generate a raceID based on the type of race (i.e. a cycling, running, or swimming race) and how many races have been run previously. The Athletes are selected from a shuffled list of Athletes. The Athlete class provides a method canRaceInGame() which determines if an athlete can race in a game. For example, an athlete that is a swimmer will not be able to compete in a cycling race. The results of each race are stored in an integer ArrayList and then sorted together with the arrayList of Athetes by the Official class. This is done using the sortRace() method, which sorts the arrayList so that it is ordered by race times. Points are given to the athletes in the top three places of the race.
4. The user's prediction is stored as an Athlete in a Game. After the results of the race are sorted, the user's predicted Athlete is compared to the Athlete at index 0 in the sorted list, that is, the winning athlete, to see if they match.

## **COSC1295 Advanced Programming Assignment 1– Sofia McKerrow and Calvin Schnierer**

Calvin Schnierer

[s3627341@student.rmit.edu.au](mailto:s3627341@student.rmit.edu.au)

Contribution: 50% of assignment

Author of classes:

- Athlete
- Database
- Game
- Participant
- RandomNumber

Although we had an outline of how to set-up the program before, it would have been better to have a more complete Class diagram and idea of the program before starting. But speaking for myself, and knowing that I can't quite do that yet, I'd say the final product was good. And the structure of it will inform how I structure things in the future.

I really enjoyed working in a group where we could help each other out, and notice errors and bugs that we'd made. It made for an effective development process that wasn't just scrambling at the last minute to get something that works. We went over things, tried to refactor and fix little things whenever we could.

I wish we had dealt with dynamic composition and coupling/cohesion a little earlier in the semester, so we could have improved on any issues of design, but overall, I'm very happy with how the program turned out.

## **COSC1295 Advanced Programming Assignment 1– Sofia McKerrow and Calvin Schnierer**

Sofia McKerrow

[s3626804@student.rmit.edu.au](mailto:s3626804@student.rmit.edu.au)

Contribution: 50% of assignment

Author of classes:

- Cycle
- Cyclist
- Driver
- Official
- Olympic
- Run
- Runner
- SuperAthlete
- Swim
- Swimmer

One area I have learned more about while completing this assignment is class design and how different classes can be used to carry out specific tasks. It would have been useful to learn more about class design, particularly coupling and cohesion, before we had already decided on how our classes would be designed. If I were to redo the class design for this project I would probably make more of an effort to reduce coupling and make some of our classes more cohesive. Good class design is something I will make more of an effort to implement for future assignments/projects.

We did encounter some errors and bugs while we were testing our program, such as errors appearing if a user tried to run a race before a game had been selected, but were able to fix the ones we found. It definitely helped having someone else to test the program and to look out for potential errors or bugs that I may have missed.