# FIT1051 Assessment Three

**Submission deadline:** Thursday 16th May 11:55pm AEST (11:55pm MYT) via Moodle

**Weight:** This assessment is worth 25% of the unit total. Submission of your code is worth 50% of this total, and your interview component (held later in Week 12) is worth 50%.

**Late Penalty:** 10% mark deduction per day

**Instructions:** Below are the coding tasks that you need to complete for Assessment 3. Your work and your submission should be independent. Please download the IntelliJ project folder below and unzip it. This will provide you with partial code in which to program your answers. Please complete each task in the appropriate section of the partial code, and submit to Moodle upon completion.

The programming portion of this assessment contains 50 marks and is comprised of the following components:

- Code correctness is worth 45 marks:
    - Task 1 is worth 16 marks
    - Task 2 is worth 4 marks
    - Task 3 is worth 10 marks
    - Task 4 is worth 15 marks
- Writing good quality code that adheres to the **FIT1051 Coding Standards** is worth 5 marks.

**Academic Integrity:** Please be reminded of the academic integrity standards that are expected of you at Monash, which were mentioned in Week 1. You should code alone and ask the unit staff for help if needed. Do not post your code in public forums or send your code to anyone. Do not copy/paste code from other sources and present it as your own — this includes use of generative AI tools. Breaching these academic integrity requirements can incur serious penalties.

**Scenario**

It's 2024, and time for another Olympics! This year in Paris, a number of countries will compete in the Women's Football (i.e soccer) event, including Australia's national team, The Matildas. You are to write a program that simulates a knock-out competition to see what country wins the Gold Medal. As the final teams are not determined (and to make your program slightly easier), 8 teams are to be in our tournament. They are (in rank order):

**Spain, USA, France, Japan, Canada, Brazil, Australia** and **Colombia.**

Each team in the simulation has a country name and a ranking, and running tallies of how many goals they have scored, and how many have been scored against them. Each game in the simulation is played by two teams, each of whom have a current score in that game.

Your program will simulate the tournament between different pairs of teams until a winner is determined. At the end of each match, the final score will be shown, including what minute the goals were scored. The team that loses is knocked out, and the winner keeps going.

The first round of games is based on ranking where highest plays lowest, so #1 plays #8, #2 plays #7, #3 plays #6 and #4 plays #5. Four teams will be left for the second round. Teams are re-ranked after the first round, where the winner of game one becomes the new #1, the winner of game two is now #2, and so on. The second round is then like the first, where highest ranked plays lowest ranked. The final round is the final of the two remaining teams.

So, for example with the countries in this rank order (1) Spain, (2) USA, (3) France, (4) Japan, (5) Canada, (6) Brazil, (7) Australia and (8) Colombia:

*Round 1: Spain v Colombia, USA v Australia, France v Brazil, and Japan v Canada*.

If (1) Spain, (7) Australia, (3) France and (4) Japan win, the next round will be:

*Round Two: Spain v Japan and Australia v France*

If Spain and Australia win, then those two teams play off in the final.

## Task 1 - Application Design (16 marks)

Download the Assessment3 IntelliJ project folder linked above, and open the project in IntelliJ. You will see that it contains a single empty class, GameDriver. This class will contain the main method of your Java application, which will run the game simulation.

Use the principles of good software design you have learned about to decide on two other class files to represent the key entities in the above scenario, so that your Java application contains three classes in total. Add these two additional class files to your IntelliJ project, giving them meaningful names. For each of these two classes:

- Identify and code the fields they will need.
- Write default and non-default constructors.
- Write getter and setter methods for the fields.
- Write a toString method.

## Task 2 - Constants (4 marks)

A team's name should not be able to be changed once it has been set.

Any game has two teams that are playing in it, and once these two teams have been set, they cannot be changed.

Use what you have learned about constants in Java to amend your code where needed, so that the implementation of the two points above is ensured.

## Task 3 - Game play simulation (10 marks)

Implement a method called playGame() in the appropriate class in your Java application, which simulates game play, updating scores until the time is up, then updating team data.

The instructions for playing a game are as follows:

- Teams score goals, each effectively worth 1 point
- Games run for 90 minutes
- For every 5 minutes that passes during game play, a team has a chance to score. The probability of scoring is relative to their ranking from the list above. This means a higher ranked team is more likely to score than a lower ranked team. The percentage chance of scoring every minute is calculated using the following formula: 20 – rankingValue

- So, for example, if Spain is ranked #1, has a 20-1 = 19% chance of scoring every 5 minutes. If Colombia is ranked #8, has a 20-8 = 12% chance of scoring every 5 minutes. (Hint: to calculate the probability of event, generate a random number from 1 to 100. There is a 1% chance of each of these numbers being generated - so, for example with Spain, you can nominate the numbers 1–19 for scoring a goal, or for Colombia you would use the occurrence of the random numbers 1-12).
- If the scores are level after 90 minutes, a penalty shootout takes place. For this you simply generate a random number for each team, and the team with the highest number wins the penalty shootout and the match.
- Once a game has finished, the final scores of each team are printed to the console, and the total goals tally for each team is updated.

In designing your method, you should apply the design principles you have learned about where relevant (e.g. modularisation, abstraction, encapsulation, high cohesion, low coupling) to ensure good code quality.

### Task 4 - Driver class (15 marks)

In the GameDriver class, write a main method which does the following:

- Creates the 8 teams, based on the country list above.
- Simulates the tournament, based on the rules above.
- Once all games have run, the name of the winner should be displayed (Gold Medal winner), along with the loser (Silver Medal).

Again, you should apply the principles of good software design you have learned about when implementing this code.

### Adherence to coding standards (5 marks)

All code written should adhere to the guidelines set out in the FIT1051 Coding Standards.

**Submission Instructions:** Please submit your IntelliJ project folder as a .zip file and submit to the Assessment 3 link on the Moodle Assessments page as shown below. If you are not sure how to zip your project, please refer to the video here. MAKE SURE YOU DOWNLOAD FROM MOODLE AFTER THAT TO CHECK IT IS THE RIGHT SUBMISSION!

**Interview component:** You will be asked to demonstrate your program at an interview in Week 12, following the code submission date. You will need to book an interview time with your tutor during your Applied Session in Week 11.

The interview will be 10 minutes in length, and consist of 8 questions. You may be asked to explain your code, your program designs, to modify your code, to discuss your coding decisions, or to explain the any of the coding concepts taught in Weeks 1-10 that this Assessment covers.

Interviews will take place online via Zoom. You must have access to a stable internet connection and a working webcam, and your webcam must be switched on for the duration of the interview. Interviews will be recorded for marking integrity purposes, and recordings will be deleted at the end of semester.

It is your responsibility to make yourself available for an interview time during Week 12, and to attend your interview on time. The interview is worth 50% of your assessment mark, and any student who does not attend an interview will receive a fail grade for the assignment. Your interview must take place before the end of Week 12.