# ILP Coursework 1: comments and marks

pbj@inf.ed.ac.uk <pbj@inf.ed.ac.uk>
Fri 01/11/2019 11:39

**To:** LIU Keyi <s1703084@sms.ed.ac.uk>
**Cc:** JACKSON Paul <Paul.Jackson@ed.ac.uk>; GILMORE Stephen <stg@staffmail.ed.ac.uk>

This report is for student s1703084

The raw overall mark is (20/25)

The awarded mark includes a deduction for any days late
(5% per calendar day).

Days late: (0)
The awarded mark is (20/25)

Overall comments:

In general this is an excellent submission which meets the requirements of Coursework 1 very well and provides a very good basis for continuing with Coursework 2.

As described in the coursework instructions, marks are awarded for 3 aspects: correctness, readability and efficiency. See below for comments on these and other aspects of your submission.

1. Correctness (15 marks)
---
Your nextPosition() method was expected be a correct implementation of the "next position" function for the drone, giving the correct next position for all sixteen possible compass directions. Your inPlayArea() was expected to correctly check whether a position is in the defined playing area. Correctness for both methods was assessed by running the provided 30 JUnit tests on your submitted code.

All tests passed.

2. Readability (5 marks)
---
Your method definitions were expected to be clear and concise. Your overall implementation of the Direction and Position classes was expected to be readable and clear.

You chose an enum to represent the Direction class, which is a concise way to express in Java a collection of constant values.
Your Position class is concise and clear with the nextPosition method being easy to understand.
Otherwise, these classes are concise, clear and readable, as required for this coursework.

3. Efficiency (5 marks)
---
The instructions said:

> The nextPosition() method will be executed multiple times in the PowerGrab game, and while testing the game, so it is important that

it should be efficient. The game development team have not yet decided on the Java compiler which will be used to compile the release version of the game, so the compiler optimisations which will be applied are not yet known. The safest assumption in this setting is to assume that the compiler will not optimise your function at all and that you should explicitly code any optimisations which can be made to the nextPosition() method.

The most significant optimisation which can be applied to the function is to avoid the repeated evaluation of constant expressions.
However on many calls to nextPosition() you recompute the same amounts by which latitude and longitude are shifted. It would have been better to define new constants to capture these repeated shift amounts and enable each to be calculated just once. This was a missed opportunity.

4. Submission directory structure
---
Some submissions had files in unexpected directories. If this was the case, some comments on this are made here. Otherwise this section is blank. No marks were deducted if there were issues.

5. Compilation issues
---
A few submissions required some manual intervention in order to get them to compile. If this was the case, some comments on this are made here. Otherwise this section is blank. All issues were very minor and in no cases were marks deducted.

END OF REPORT

--