

ECM1410 Object-Oriented Programming

Development paradigm in summative coursework

Diogo Pacheco

February 1, 2022

Pair programming is a popular development approach, primarily associated with *agile* software development, and is used across the software industry. In pair programming, two software developers work together, at the same time, on a single machine, to generate the solution to a given problem. Normally, the pair would be side-by-side in the same physical machine, but given the current restrictions, we may need to work remotely in the same machine by sharing our screens. One (the *driver*) physically writes the code while the other (the *observer*) reviews each line of code as it is generated. During the development, the roles are switched between the two programmers *frequently*. The aim of the split role is for the two programmers to concern themselves with different aspects of the software being developed, with the observer considering the strategic direction of the work (how it fits with the whole, and the deliverables), and the driver principally focused on tactical aspects of the current task at hand (block, method, class, etc.), as well as allowing useful discussion between the developers regarding different possible solutions and design approaches.

Research into pair programming has indicated that it leads to fewer bugs and more concise (i.e., shorter) programs than development by a single programmer, and achieves this in a shorter period of time (albeit with slightly more total developer time, due to two developers being involved at once).¹ Additionally, it also facilitates knowledge sharing between developers, which can be crucial for a software house, and is important for the continued learning of the developers involved. Often in industry pairs are generated by cycling through developers from a larger team, so everyone is eventually paired with everyone else at some point.

This assignment will introduce you to practical paired programming, through the development of a Java package. As such you will need to form pairs.

Students must inform Dr Pacheco of pair membership by 5pm on Wednesday 9th February (one email per pair, but with both students in copy). The subject line of the email must be “ECM1410: partner selection”. If you cannot find a partner and wish to be placed in a pair with another student, please attend the lecture on Thursday 10th February as we will draw additional pairs. The pairs formed during the lecture should also send an email, as described above, immediately after the session.

Important points to note

- The coursework can only be done in pairs or solo. No exceptions for groups bigger than two. The task will be designed such it can be performed by a single person within the time given. However, if you want, you can benefit by practicing pair programming and, consequentially, delivering a better code in a shorter period of time.

¹A. Cockburn & L. Williams (2000). “The Costs and Benefits of Pair Programming”. Proceedings of the First International Conference on Extreme Programming and Flexible Processes in Software Engineering (XP2000).

- A cover page which details how you would like the final mark to be allocated to each of the developers, based upon your agreed input, will be required in the submitted coursework. If both parties took equal roles, this would be 50:50 (which is what the vast majority of submissions typically choose, and the general expectation for this assignment). You might alternatively choose, for instance, 53:47 if you both agree that one party may have contributed a bit more than the other. However, in pair programming both the driver and observer roles are vital, and should be switched frequently between developers, and so both members should contribute roughly equally. The maximum divergence that may be indicated on the cover-page is 60:40.
- Pair programming is categorically not two developers working separately on two different machine. Side-by-side (virtual, sharing screen) communication developing on a single machine is a key aspect of the approach.
- **The module lecturing team reserves the right to split pairs where one student is not engaging with the coursework. The coordinator also reserves the right to assign non-contributing students a mark of 0.** In the rare situation that you are paired with a student who is not contributing (e.g. not replying to emails and/or not meeting up for pair-programming sessions) you must inform Dr Pacheco of the situation **within one week of release of the coursework specification** to facilitate the aforementioned splitting of pairs if necessary.

Given the above process and timelines, please ensure that you arrange to meet up and start the work as soon as the coursework specification is released to reassure yourself that you are partnered with a student who wants to actively contribute to the coursework. It is an expectation that pairs have *at a minimum* two pair-programming sessions in the first week of release of the coursework.

- It is not permitted for students working on the pair programming assignment to collaborate on the assignment with any other student *apart* from their named partner. Those doing so will be subject to academic misconduct regulations and penalties. Please refer to the undergraduate handbook for details on collusion, plagiarism, etc.

The table below gives example coursework reweightings based on different ratios:

Ratios and adjusted marks					
Raw mark	60:40	58:42	55:45	52:48	50:50
20	24, 16	23, 17	22, 18	21, 19	20, 20
30	36, 24	35, 25	33, 27	31, 29	30, 30
40	48, 32	46, 34	44, 36	42, 38	40, 40
50	60, 40	58, 42	55, 45	52, 48	50, 50
60	72, 48	70, 50	66, 54	62, 58	60, 60
70	84, 56	81, 59	77, 63	73, 67	70, 70
80	96, 64	93, 67	88, 72	83, 77	80, 80
90	100,80	100, 80	99, 81	94, 87	90, 90

Note, marks > 100 are capped at 100 and the excess redistributed.