

Refactoring Golf

based on a workshop run by Dave Cleal, Ivan Moore and Mike Hill

Import the project into IntelliJ. You will see a set of “before” and “after” code examples, as `hole(N) -> hole(N+1)`.

The aim is to refactor the “before” code into the “after” code, getting from start to finish with as few refactoring moves as possible.

Each move (or keypress) that you make has a certain cost - see the table below. Add up your “golf score” as you make the changes. In refactoring the code you should aim to transform it in ways that do not affect its behaviour. Unsafe moves (that don’t preserve the code’s behaviour) have a higher cost.

We’ll work on each example in turn. Once we’ve got a solution, the group with the lowest score will present to the group, then we’ll try another example.

Scoring:

Operation	Cost
Selecting Text	Free
Navigating Around	Free
Reformatting	Free
IDE Refactoring Operation	1 point
Editing Operations	1 point per keystroke (key combinations such as ctrl-c to copy count for 1 point)
Any move that changes behaviour or stops compilation	2 points
Subsequent moves while code is still broken	additional 2 points