

# Informatics 1: Object Oriented Programming

## Tutorial 05

### Week 6: Peer Evaluation


Brian Mitchell (brian.x.mitchell@ed.ac.uk)

Vidminas Vizgirda (s1750767@ed.ac.uk)

## 1 Introduction

This tutorial is about processing the feedback from your Assignment 2 Part 1 submission to improve your work, preparing for the Part 2 submission.

You will look at your individual and general group feedback and have opportunities to work with other students evaluating and discussing each others' designs.

 If you can, read your tutor's individual feedback on your Assignment 2 Part 1 submission (if any) before the tutorial starts. It will help you make better use of tutorial time.

The main outcomes from this tutorial are making refinements to your work and sharing suggestions with other students. The coursework is a community exercise not a competition – by giving useful feedback to others, you will likely receive useful feedback for yourself too.

## 2 Tutor feedback

The tutors in your session will present group-level feedback. This will vary by tutor but it mainly covers points that apply to multiple students or even all of you. Listen to the group feedback carefully.

## 3 Starting Assignment 2 Part 2

In Assignment 2 Part 1, you chose a Code Golf challenge to focus on. For Part 2, you will need to improve your submission from Part 1, taking into account tutor and peer feedback. The rest of the tutorial will be dedicated to working in small groups on starting Part 2.

### Task 1 - Find challenge partners (5 minutes)

◀ Task

Try to find a partner (not necessarily from your regular tutorial table) who has chosen to focus on the same Code Golf challenge as you. It might not be possible for everyone to find someone else with the same challenge. If that happens, you can join groups of 3 or pair up with someone working on another challenge.

You now have a few minutes to discuss the feedback you received for your submission (if any). Whether or not you received any personal feedback, also think about how the group feedback applies to your case or why it does not.

## Task 2 - Briefly discuss challenge-specific feedback (5–10 minutes)

◀ Task

If you don't already know the person you are working with, introduce yourself.

You now have a few minutes to discuss the feedback you received for your submission (if any) and the group feedback.

Please do share anything that you found unexpected or helpful: although feedback is student-specific, sharing advice is useful to everyone and good citizenship. Be fair with your partner(s) and ensure you have equal chance to discuss each other's work.

## Task 3 - Prepare the Assignment 2 Part 2 project (5–10 minutes)

◀ Task

Although we are working in pairs, unlike in previous tutorials, we will not be pair programming. Instead, everyone should prepare their own project, if possible. If not, you can share computers and work together with someone else on their project, but take notes about the feedback you receive, as you will need to have a record of it.

1. If you haven't already, install the [Open-Dyslexic fonts](#) in case someone you work with during this tutorial needs them.
2. Follow the setup instructions on Learn: [Inf1B](#) > [Assessment](#) > [Assignment 2 Part 2](#) > [Tasks and assignment files](#) to prepare your Assignment 2 Part 2 project. This will involve downloading a new project and copying some files over from your Assignment 2 Part 1 project. Make sure you copy the files instead of overwriting the original ones, as you may need the original copies to show side-by-side what you have improved.
3. Open Part 2 in IntelliJ, remembering to open the directory itself not one of the files inside the directory.
4. In `acknowledgements.md` add your pair partner's name with a note saying they are a student who helped you during the tutorial.

## Task 4 - Adjust the description in `worksheet.md` (5 minutes)

◀ Task

Your tutor will now give you time to make adjustments to your worksheet based on the feedback. This is done in stages starting with the description in `worksheet.md`.

**Clearly state at the start of your description in `worksheet.md` what level of beginner your worksheet is aimed at** — if you're not sure: guess. You'll refine this target audience ready for the assignment submission, so don't worry about it too much now. But the sooner you can settle on a target audience, the sooner you can focus your worksheet to that particular group.

Concentrate first on the explanation of the problem in `worksheet.md`. You won't necessarily have time to make all the changes recommended so make notes for anything you don't have time for.

## Task 5 - Export your `worksheet.md` to a PDF (2 minutes)

◀ Task

Use IntelliJ to convert your worksheet from Markdown into a PDF. **Make sure you have the preview or editor+preview open** first, then in the top menu select: [Tools](#) > [Markdown Converter](#) > [Export Markdown File To...](#), choose PDF as the format and specify the filename and location you want.

If you have any trouble, see the documentation here: [www.jetbrains.com/help/idea/markdown.html#convert-markdown-files](http://www.jetbrains.com/help/idea/markdown.html#convert-markdown-files)

## Task 6 - Adjust the design in `src/Design.java` (5 minutes)

◀ Task

Now look at `src/Design.java` and apply any changes recommended by your tutor. You might not have time to make all the changes recommended but do what you can and make notes for anything you don't have time for.

## 4 Peer evaluation round 1



Figure 1: Photo by [R. Nyberg, USAID](#) on [Pixnio](#). Available under [CC0 licence](#).

Although there hasn't yet been time to produce a "perfect" worksheet, you are now going to test it with your partner. Getting feedback early can actually help you save time by avoiding spending time on refining something that you might decide to not keep at all.

We only have limited time in the tutorial, so you will need to be prepared to work quickly. First, read all the tasks below so you know what to do.

You will have 15 minutes for the peer evaluation round, so don't get hung up on any one step too long – if you get stuck, ask a tutor for help or skip ahead.

### Task 1 - Decide roles

◁ Task

In your pair, decide which one of you is going to be the **tester** and who is going to be the **observer**. You will swap roles later. If you are working in a group of three, pick two observers and one tester.

### Task 2 - Read `worksheet.pdf` (3 minutes)

◁ Task

The tester now has **3 minutes only** to read the observer's `worksheet.pdf`. If the tester has difficulty reading then arrange with the observer how to proceed, for example reconfiguring the markdown's CSS and re-exporting or reading the text aloud.


Remember the worksheet is a work in progress. If there are any changes that have not yet been made, the observer can outline them briefly at this stage but there isn't time for much detail. The tester is not allowed to ask any questions yet but will have the chance soon.

While the tester reads, the observer watches and make notes. In particular observe the effect your draft worksheet has on the tester, for example do they look confused or interested. Don't worry if you're not good at reading others' emotions: the tester will tell you later. The tester is not allowed to ask any questions at this stage, so if they do, politely refuse: questions will be answered shortly.

### Task 3 - Question `worksheet.pdf` (1 minute)

◁ Task

After the 3 minutes, the tester has **1 minute only** to ask for clarifications and the observer must try to answer succinctly. The tester should also commend parts that are easily understood or encouraging, or identify parts that were confusing. Both of you must bear in mind this is a work in progress and everyone has different experience of programming and differing ability to express themselves.

 Always assume that any confusion is caused by the writer having not been clear enough for that particular audience. Confusion is no one's "fault." Communication and discussing designs

of code are essential software development skills, and you can improve them through peer evaluation like this.

#### **Task 4 - Read `src/Design.java` and/or review the design diagram (3 minutes)**

◀ Task

The tester now has **3 minutes** to read `src/Design.java` and/or review the design diagram (the observer chooses what would be most helpful), and as last time, without asking questions. If there are any planned changes that have not yet been made, the observer can explain them briefly at this stage. The observer may use this time to make notes about the worksheet based on the questions and comments. If the tester is using your computer, you can't edit the files while they are working because that would be distracting.

#### **Task 5 - Question `src/Design.java` and/or the design diagram (1 minute)**

◀ Task

After reading the design document, the tester may ask for clarifications and make comments on the design for **1 minute**. Again, remember this is a work in progress and that everyone has different levels of experience in programming generally and Java specifically. Different designs are also possible and one person's way of answering the challenge may not be another's. You can discuss different approaches later.

#### **Task 6 - Think about the solution (2 minutes)**

◀ Task

The tester now has **2 minutes** to think about how the student who has set the worksheet (the observer) wants them to solve the challenge. It might be useful to think out loud (say what you are thinking) so that the observer can understand. This is also good practice for job interviews where you are given a problem to solve.

The observer spends these 2 minutes making notes about the design based on the feedback from the tester.

#### **Task 7 - Discuss the solution (5 minutes)**

◀ Task

You have **5 minutes** to discuss the instructions, design, and the solution(s) that the worksheet is described. Be kind and be generous. The tester leads this discussion by saying what they think the instructions and design wanted them to do and what they thought of this. This can be in terms of the directions, for example which parts of the instructions were (not) clear and why were they (not). It can also be in terms of design, for example whether part of a design could be simplified or moved to make the overall design clearer. It's perfectly fine, healthy in fact, if the tester and observer have differing ideas about how the challenge can be solved. Both of you should be prepared to change your minds about what design is "best" suited to the target audience or whether the target audience should be changed to suit the design, for example if the design is a "good" one but needs knowledge from after Week 5 of the course.

## **5 Peer evaluation rounds 2-4**

Once 15 minutes are up, move on to the next peer evaluation round, even if you haven't finished everything from the previous one. It's better for everyone to get some feedback on part of their work than for only a few people to receive feedback on all of theirs.

#### **Task 1 - Swap roles and repeat (15 minutes)**

◀ Task

Swap roles between tester and observer and repeat the tasks from peer evaluation round 1.

#### **Task 2 - Peer evaluation round 3 (15 minutes)**

◀ Task

If you worked in a group of 3, swap roles again, so that everyone has a chance to be the tester and everyone's project receives feedback.

Otherwise, you may either finish up any leftover work from previous rounds or find a new partner, ideally someone doing a different challenge to you and a different challenge from any you have tested so far. Don't forget to add their name to your `acknowledgements.md`.

### Task 3 - Peer evaluation round 4 (remaining time, if any)

◀ Task

Depending on how much time is left in the tutorial, you may switch roles and repeat another peer evaluation round or you may choose to write some notes.

There may not be time for this in all tutorials. If that is the case, that's okay – everyone should have had the chance to receive feedback on their project at least once.

## 6 Memos



Figure 2: “Person Writing on Pink Sticky Notes” by Bruno Bueno. Free to use under the Pexels licence.

Make notes about what you remember from today's session as soon as you can after the tutorial. Some things might seem obvious now and it might seem like you will remember it all for sure, but for most people, it's very easy to forget feedback soon after a review session. To avoid this, it's a good idea to write notes down while it's still fresh in your memory.

Notes like this are called “memorandums” or “memos” and are common in user research.

Finishing all the improvements on your own is the rest of Assignment 2 Part 2. Good luck!