

Project Phase 2

Last modified: Sunday 8 March, 2020

With object oriented programming, we aim to make our code: easy to work with, easy to maintain, and easy to extend. To see how close you have come to these goals, you will extend your program to include more functionality. Some of the extra functionality is mandatory. The rest you can choose from the list, or invent yourself. If you are concerned that your extra feature may be too complicated or too simple, please ask Lindsey, in person, at office hours, or before/after lecture. For weeks 9, 10, and 11, her Thursday office hour will be in WW 107 D from 6-7:30 pm or until all the questions have been answered. Also, she will stay after every lecture for at least 30 minutes to answer questions about the project.

If you have a group of four people or fewer, you can add extra members for Phase 2. This will be sorted out during lab time on Tuesday March 10.

Getting Started

There is a phase2 folder in your repository. You will see it, the next time you git pull from MarkUs. One group member should re-clone the repo with phase2 as the top folder and copy-paste your entire project from the phase1 folder into the phase2 folder and git add/commit/push so that everyone else can re-clone the repo with phase2 as the top folder and pull the files.

Alternatively, if you want to change your packaging or add new packages, you can have one team member create a new project inside the phase2 folder, then create the folders and copy-paste your java files into them.

New Features

Your final project will have the following features:

1. All functionality from Phase 1 should be implemented and running properly.
2. There can be multiple calendars per user.
3. Multiple users can have the same event in their calendar. How you get one user to create the event and then add the other user is up to you. (For a way that involves extra functionality beyond just "adding" the second user to the event, see the optional features below)
4. The ability for the user to delete/edit events, alerts, and memos.
5. You should be able to demonstrate that all of your features work in a limited amount of time, including features that require multiple days to go by.
6. At least one feature that is specific to your team.

Optional Features

Your program should also include three of the following features:

- Implement a graphic user interface (GUI). This counts as two features.
- messaging capabilities, where users can invite each other to an event with an attached message and the other person can accept or decline with the option to reply to the message
- Give the user the option to open their default browser and talk about their event on social media. This could involve files (example: pictures, video) or you can implement it with text only. At least two social media platforms should be accessible this way.
- A user can postpone/reschedule an event and also duplicate it -- in other words, easily recreate the event at additional time(s)/day(s). Lastly, the user can postpone the event indefinitely. In other words, there should be a way for an event to persist without a date until the user gives it one.
- Implement a "To do" list that allows items to be associated with events. Unlike a memo, it is the individual items in the list that are associated with the events. (A memo can be associated with an event, but not a specific line in the memo.) Multiple "To do" lists can be associated with the same events or different events.
- Allow users to view their schedule by the hour, day, week, or month. It should be possible to use this feature to find out on which day of the week will be the 1st of May, 2055. It should also be possible to schedule holiday events like Canada Day (which is always on the first day of July) or Labour Day (which is always the first Monday in May).

What You Should Submit

Your program should compile and run on at least two laptop computers owned by your teammates. This is to facilitate the Phase 2 presentation, which you will do on Tuesday March 31, Wednesday April 1, or Thursday April 2. More details about the presentation will be released before your last lab which is on Tuesday March 24.

You should submit to MarkUs, in your Phase 2 folder:

- Your fully functional program, complete with Javadoc for every class, public variable, and public method.
- A UML diagram of your entire program in a file called design.pdf or multiple uml diagrams for different parts of your program in files called design1.pdf, design2.pdf, etc.
- Any configuration files that are required to run your program.
- A file called README.txt that contains any instructions we might need, if we want to run your program later.