



COMPGC01 Introductory Programming Coursework

Due date: 15 December 2017, 23:55

Feedback returned: One month after demos running in weeks 21-24

Moodle Submission Only

COMPGC01 will assess your ability to demonstrate your programming knowledge, purely in Java. It does not require a fully completed application to get a successful mark. However, the more completed it is, the higher your marks will be as you will see from the mark scheme below.

Since it's an introductory course, and in order to enforce the learning process, we encourage pair programming. See definition and benefits [here](#).

For the current coursework, you will work ***in pairs*** on a cinema booking management system.

- You are responsible for choosing the student to pair with. It can be a student from any of the lab groups A, B, etc.
- You need to give your teaching assistant the names of the two students in your group anytime during practical sessions but no later than **November, 3rd 2017**.
- After the deadline above, students with no teammates will be grouped together into pairs.
- No swapping is accepted after November, 3rd 2017 and no complaint about a student not contributing enough will be taken into account.

Cinema Booking System

You are required to design and implement a cinema booking management system with the following minimum requirements:

1. The management system should be implemented as a JavaFX GUI application using buttons, tables, listeners, etc.
2. The system handles two different profiles; the cinema employee and ordinary customers. Both types of users need to login before being able to use the management system
Note: Don't over complicate the registration process. You can simply check the user's credentials by reading a manually created file containing username/password pairs.
3. Since, there are two types of users, your application should display a different view depending on the logged-in user (you can use tabs on the same application to allow different profiles to login or display the right view after



login process is successful. This can be done by, detecting the role of the logged-in user on the fly).

Cinema employee profile

4. The cinema employee should be able to login to the cinema management system using a username/password pair.
5. The cinema employee can add films with their respective dates and screening times. A film is represented by a title, a small image, and a brief description. You can use dummy images and text or copy some from the IMDB.com website.
6. For a given film/date/time, your application should provide an up-to-date graphical representation of the cinema room setting. It should graphically distinguish between booked seats and available ones (consider using different seat icons). It should also show labels (e.g G13) and approximate position of the seats within the cinema.
7. The employee view also shows basic information about the total number of seats, booked ones and available ones for a given film/date/time.
8. The employee can export a list of films (i.e. titles), dates, times and number of booked and available seats. The list is a comma separated values text file.
9. The cinema employee can logout from the application.

Cinema customer profile

10. After login, a customer should be able to update his/her profile to use for future bookings. The profile is composed of basic information such as surname, first name, email address. There is also booking history that is updated automatically when the user makes a booking or delete an existing one.
11. He/she can pick up a date, get a list of films available on the selected date with their respective information and available screening times.
12. The customer can book a seat (for a given film/date/time) by clicking on the graphical representation of the seat. If the seat is available, the seat icon changes to 'booked'. If the seat is not available, an error message is displayed.
13. When the customer clicks on the 'Confirm booking' button, a summary of his booking is displayed and the booking added to his history.
14. A customer can delete an existing booking from his list if it is a future booking only, otherwise an error message is displayed.
15. The cinema customer can logout from the application.



Note: Customer's information, bookings, etc., as well as employee films settings and room bookings should be persistent across login and logout sessions.

Keep it simple!

(*) The cinema contains one single screening room. (*) a film lasts exactly one hour. So you just need to be careful when 'manually' creating films/dates/times in step 6. Make sure screening times don't overlap.

Deliverables

The grade for your GC01 coursework will depend on the quality and correctness of your programming implementation. You are required to submit a single ZIP file on moodle containing:

1. The source code files of your project, including the Eclipse project workspace (should be able to compile as is)
2. The class-generated files of your project in JAR format. The jar file should be executable.
3. Javadoc appendix.
4. A text file with a **link** to your UCL MediaCentral (<https://mediacentral.ucl.ac.uk/>) video demonstrating all the features of your application. You can add voice-over or text comments.

Note: Before submission, make sure the maximum size of your zip file doesn't exceed 50MB.

Demonstration of your project

For the sake of providing an accurate, and transparent marking of your work. All students pairs will need to demo their project.

1. On your timetable, there is a set of booked slots in weeks 21-24 (term 2). There is no lecture or practical. You don't need to attend them all. They are booked for projects demos.
2. Each pair will be given 15 minutes (approximately) on one of the slots. To demonstrate its project and answer assessors' questions.
3. The students who request it can be given a 'preliminary' feedback on their work at the end of the demo. ***The final mark, however, also depends on the actual testing of your application by us offline.***
4. Both teammates need to be present during the demo.
5. The schedule of the demonstrations for the individual pairs will be published just before the beginning of the demo weeks.
6. All projects will be assessed by at least two assessors
7. All projects will run on one of our machines not on your laptop. So, please do not make assumptions about configuration. Your submitted project should be self-contained and shouldn't require any configuration on our side to run. The



only assumption is that we provide a machine with Eclipse, Java 8 and its standard libraries.

Mark scheme

In both options, you will be assessed clearly on the following, which must be shown in a useful context. *Please note that all non-optional requirements must be implemented for a 70%:*

1. Has a fully implemented JavaFX Graphical Interface.
2. All requested features implemented
3. Has appropriate Java objects and demonstrates inheritance.
4. Java exceptions are used and code held in Java Packages.
5. JavaDoc comments generated and a JavaDoc folder submitted.
6. **Very important:** Make sure your code is robust enough by testing it before submission, as you may lose marks if your software application raises errors or behaves strangely during the demo.

Distinction marks

UCL teaching policy promotes students' innovation and creativity. In the current context, your application can include any additional feature you might think your software should have and not listed above. Your final mark will therefore depend on the innovative aspects you add to your software.

A detailed marking sheet is available at the end of this document for your convenience.

Notes

- This coursework is compulsory.
- If you include the use of 3rd party libraries in your solution (which is welcomed so long as it is cited in the comments in that Java source file) ensure any cited works are free and open for reuse.

Tips

- Start this coursework immediately, so not to get behind with other coursework deadlines.
- Revise eclipse strategies for creating packages, compiling, debugging methods, add breakpoints, how to load existing Eclipse projects and how to import other libraries.
- Refer to the 'useful links' section below for some additional tips.



Reminder

Plagiarism of any kind, on the applications and on your content prepared will not be tolerated! All of the software developed must be your own works. Any code examples used from online sources and tutorials must be CITED. Failure to do so will account to Plagiarism, which will be dealt with under the appropriate Examination Boards.

Useful links

1. ***Up and Running with Eclipse*** with Charles Kelly. Available from the UCL Lynda website <https://www.ucl.ac.uk/lynda>
2. ***User Interface Design For Programmers*** by Joel Spolsky. A very long article on how to design intuitive user interfaces. **Tip:** The last paragraph summarizes the main tips: <http://www.joelonsoftware.com/uibook/fog0000000249.html>
3. ***Object-oriented design principles and the 5 ways of creating SOLID applications*** by Kaur Matas. <http://zeroturnaround.com/rebellabs/object-oriented-design-principles-and-the-5-ways-of-creating-solid-applications/>
4. **Extenuating circumstances & late submissions:** Please consult your academic manual available here <https://www.ucl.ac.uk/srs/academic-manual/c4/failure/late-submission>



COMPGC01 marking sheet (2017/18)

Description	Description	Marks
Project structure	This rubric marks your project structure: jar file, video, use of packages, inheritance exceptions, etc	10
Cinema manager view features	This rubric assesses the implemented manager features as described in the coursework sheet.	30
Cinema customer view features	This rubric assesses the implemented customer features as described in the coursework sheet.	20
Data persistence across login sessions	This rubric assesses how you implemented the data persistence.	5
Graphical user interface	this is an overall evaluation of your GUI. See reference in section 'Useful links' in the coursework sheet.	5
Extra features	You will get marks for any extra 'useful' feature you implement on top of the required ones.	30
Total		100