



COMPGC01 Introductory Programming Coursework

Due date: 05 December, 2015

Feedback returned: 12 January, 2016

Moodle Submission Only

Important Note!

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. The coursework deadline is 05/12/2015.

You have two project options on this COMPGC01 coursework to choose from:

1. Work individually on a patient registry management system.
2. Work as a group of ***two*** on a robot simulator application.

Option 1 - A patient registry management system (individual project)

You are required to design and implement a ***simple*** patient management system with the following minimum requirements:

1. The management system should be implemented as a java GUI application using frames, tables, buttons, listeners, etc.
2. The practitioner should be able to login to the patient management system using a username/password pair.
3. Register a new patient with minimal information such as name, first name, DOB, address, emergency phone number, medical condition, appointments, billing, comments, etc. Each record might contain a profile picture of the patient and a set of pictures describing his/her condition such as brain scans (please use any sample image, It doesn't have to be realistic).
4. The field containing the medical condition data should be clickable and pointing to a wikipedia page or any URL with the description of that condition.
5. Edit an existing patient record, by editing existing information such as dates/times of appointments, uploading new images or deleting others.
6. Delete an exiting patient record. Your program should ask for confirmation before deleting a record.



7. Search the list of patients based on any of the stored fields, i.e., name, DOB, address, etc. and produce a list of corresponding records.
8. (*Optional*) Export a list of selected patient records as a comma separated file or any other format.
9. (*Optional*) Import a comma separated (or the format used in 8) file containing patients' records into the system. The new patients data will be appended to the existing ones already stored in the management system.

Note: It is NOT mandatory to use a database to store patients' data. You can store it in a text file.

OR

Option 2 - A wheeled robot simulator (group project)

In a group of *two students*, you are required to design and implement a *simple* wheeled robot simulator with the following minimum requirements:

1. The application should be implemented as a java GUI application using frames, buttons, listeners, etc.
2. The robot is represented by a rectangle, or an image (see java 2D API).
3. The robot can a) move forward or backward with a given speed b) rotate left/right with a given angle.
4. A user can control the robot through a set of GUI commands such as buttons and text fields, etc, or keyboard keys.
5. The robot can move in a maze that can be represented by a set of connected lines.
6. A text or audio alert is triggered if a move would cause the robot to collide with one of the walls (lines representing a wall).
1. The robot has an initial battery charge that decreases each time the robot moves one unit of distance. The robot stops if battery charge is equal to zero.
2. A text or audio alert is triggered when the robot charge is less than 10% of the maximal charge.
3. A text (java text area for example) is updated live with the coordinates of the robot during movement; it's angle, distance crossed and battery level.
4. Record all commands executed by the robot in a text file with a timestamp.



5. Read commands from a text file and execute them by the robot at once.
6. (**Optional**) Study differential steering of wheeled robots and provide an implementation. Here is a starting point:
<http://rosum.sourceforge.net/papers/DiffSteer/>
7. (**Optional**) The parameters corresponding to the robot geometry such as the dimension of the robot, the distance between the two wheels, the wheels radius, etc are stored in an XML file and loaded when the application starts. The implementation provided in requirement (12) should be adapted accordingly.

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.
3. A text file list of the features implemented.
4. Javadoc appendix.
5. A video demonstrating all the features of your application. You can add voice-over or text comments.
6. (**Optional**) tests folder for any other demos (compiled and source code) that stand alone in demonstrating features of your system, perhaps you had a number of prototypes as you learnt.

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 appropriate Java objects and demonstrates inheritance. (10%)
2. Makes use of Arrays, ArrayLists or other Java.Util collections where appropriate. (10%)
3. Has a fully implemented Java Graphical Interface. (40%)
4. Java exceptions are used and code held in Java Packages. (5%)
5. Javadoc comments generated and a Javadoc folder submitted. (5%)

Advanced Reading (distinction) marks-

1. Demonstrates at least 5 Java based libraries that are not in the lecture notes, being used in an appropriate and useful context. The Java libraries do not need to be from the Java SDK but they should at least be open source. (5%)
2. Implementation of optional features. (25%)



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.

You will learn along the way how best to test it with small test apps. You may keep these in a separate folder called tests if you wish to submit it with your main Zip file as supporting evidence of your project's capabilities. This will greatly help you in future programming courses.

Useful links

1. ***Up and Running with Eclipse*** with Charles Kelly. Available from the UCL Lynda website <https://www.ucl.ac.uk/lynda>
2. ***How to create jar files in Eclipse***
<https://www.cs.utexas.edu/~scottm/cs307/handouts/Eclipse%20Help/jarInEclipse.htm>
3. ***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>
4. ***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/>

~~~ End ~~~