




**School of Science**

# **COSC1295 Advanced Programming, sem 1 2021**

## **Assignment Part 1: Console Implementation**

	<b>Assessment Type: Individual assignment; no group work permitted.</b> Submit online via Canvas. Marks are awarded for meeting requirements as closely as possible according to assignment specifications and the supplied rubric. Clarifications/updates will be made via announcements/relevant discussion forums.
	<b>Due date: Multiple milestone submissions. Final submission due 9am Monday May 31, 2021.</b> No late submissions accepted unless Special Consideration is granted in advance.
	<b>Weighting:</b> 50% of your final grade. In addition, multiple milestones through the semester to a total of 15 marks.

## **1. Outline**

**NOTE:** Carefully read this document. In addition, regularly follow the Canvas assignment discussion board for assignment related clarifications and discussion.

In this assignment you are required to work individually to develop a Resident Care System using Java. Using Java as a vehicle, you are required to demonstrate your understanding of object-oriented design principles, design patterns, generics, graphical user interfaces, unit testing and object relational mapping building on the foundations laid by the Programming Fundamentals course and further developed in COSC1295. The system you develop should be maintainable and extensible as well as produce substantial value for the stakeholders by improving the level of patient care. The marks will therefore be awarded both for your functionality as well as justifying the design decisions and code quality. During the final face to face assessment you will also be required to justify your design decisions as well as explain how your program can be extended to meet other related requirements.

This assignment is structured to let you incrementally build your product throughout the semester by incorporating the OO design principles, Java constructs and technologies introduced in the notes and videos on Canvas, and developed in the lecturorial and prac sessions. You are required to ensure you make steady progress and marks will be awarded in the prac classes for completing specific milestones. The main purpose of this assignment is to develop and demonstrate your problem-solving skills by applying the concepts taught for a reasonably complex open-ended problem (and to create a complete working system based on agile specifications). You should not be overly concerned about every minute detail and are free to make reasonable assumptions (you may check with your lab-tutor if in doubt, who will also mark your final assignment).

**Disclaimer:** the specification in this assignment is intended to represent a simplified version of a system in real life and thus is not meant to be an accurate simulation of any real system or service that is being used commercially.

## 2. Assessment Criteria

This assessment will determine your ability to implement Object-Oriented Java code according to the specifications shown below. In addition to functional correctness (i.e. getting your code to work) you will also be assessed on code quality. Specifically:

- You are required to demonstrate your understanding of object-oriented programming constructs encapsulation, inheritance and polymorphism.
- You are required to promote performance and robustness by using well tested collections and by catching and handling all exceptions.
- You are required to demonstrate that you have done adequate unit testing considering various business rules using the JUnit framework.
- You are required to demonstrate all the objects stored in memory can be persisted to a file and restored when the program starts again.
- Your interaction design must anticipate what users might need next and include appropriate JavaFX elements to make the system accessible, intuitive, usable and efficient (fewer clicks/keystrokes)
- You are required to ensure all historic data can be archived to a relational database allowing access by external software and the current system.
- Your object-oriented design should provide high cohesion and low coupling following well established design principles and patterns. You should justify your design decisions that led to the final product using detailed comments in the beginning of each class.
- You should reflect on the difficulties/limitations of the technologies used and suggest how these can be improved by researching and identifying other technologies and tools in a separate one-page report.

Further specifications and assessment criteria will follow throughout the semester.

## 3. Learning Outcomes

This assessment is relevant to the following Learning Outcomes:

CL01: use the Java programming language in the implementation of small to medium sized application programs that illustrate professionally acceptable coding and performance standards.

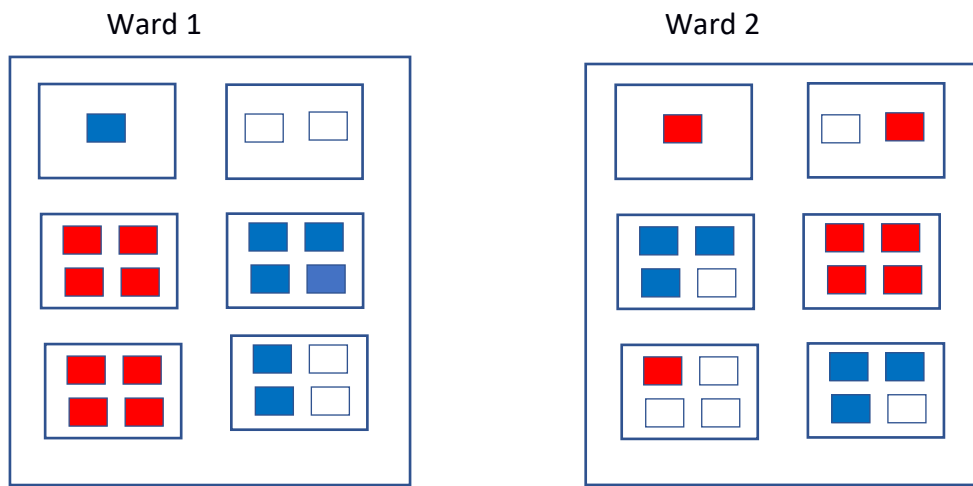
CL02: demonstrate knowledge of the basic principles of the object-oriented development process and apply this understanding to the analysis and design of solutions for small to medium scale problems.

CLO3: describe and apply basic algorithms and data structures, in particular simple searching and sorting of data stored in data structures and manipulating data.

CLO4: implement basic event-driven programming and graphical user interfaces in Java.

## 4. Product Description

You are required to develop a system to manage the high care patients for the Able Care Home. The management wants a visual interface that maps the patients to their respective beds to help nurses easily check the prescription details and administer the medications in an effective manner. Moreover, it may be necessary to change the bed assigned to an existing resident when a new resident is admitted reflecting their gender and medical condition (need for isolation). In order to comply with recent regulations, the management wants to archive all the patient related details including medicine prescription and administration details to a database that can be accessed and audited by external parties. In this assignment you are only required to deal with two wards with 6 rooms, each with the number of beds per room varying from 1 to 4 as shown below.



### Functional Requirements to be Carried out through a Graphical User Interface

- Allow manager to add new residents to one of the vacant beds
- Allow manager to add staff (doctors, nurses) or modify staff details (e.g., password) or shift days/times
- Allow medical staff to check the details of a resident in a bed
- Allow a doctor to attach a new prescription to a patient in the selected bed stating various medicines to be administered at specific doses at specified times
- Depict the gender of the resident in a bed using blue/red
- Allow a nurse to move the resident to a different bed
- Updating the details of medicines administered for a resident in the selected bed
- Any action (e.g., adding a patient to a bed, updating a prescription, administering a medication, etc) must include day+time and id of the staff member performing the action and these are recorded with the action. An Exception (different ones) should be raised if the staff member is not authorized to perform the action, or is not rostered for that day+time.

### Other Functional Requirements

- Saving all the patient related records to a file before exiting the system, allowing it to be restored (which need not be accessible to external parties).
- Archive all the patient related details including medicine prescribed and administered throughout the stay when discharging a patient (allowing them to be audited)

**Process, Quality, Security, Testing, Personal-Reflection and Technology Related Requirements**

- The program submitted should be refactored to promote code maintainability by using the object-oriented concepts including encapsulation, inheritance and polymorphism.
- Use of design patterns such as singleton and/or adherence to design principles such as SOLID.
- Well-designed test cases showing main business rules are implemented. For example, an attempt to add a new resident to a vacant bed should succeed while an attempt to add a resident to an already occupied bed should fail with appropriate diagnostic messages.
- Catching and handling all error conditions to create a robust system.
- A one-page document justifying all the design changes and lessons learned.
- Ensure only doctors can add prescriptions and only nurses can administer medication. Only manager with own username/password can add username/password for doctors and nurses.

**Progress Milestones**

You will be required to demo your progress in the lab regularly to demonstrate progress.

The specific milestone requirements and associated dates will be communicated via Canvas.

## 5. Project Phase 1

**Phase 1 (Focusing on Class and Interface Design, Use of Collections, JUnit Test cases, Serialization and Exceptions)****Due: Start of Week 7**

Implement the staff and location related functionality incorporating collections, generics and exception handling. As well as the rules for allocating patients to beds and recording medication, etc, requirements, you must account for the following rules/compliance on care staff managing patients:

- Your program should ensure nurses are assigned to two shifts from 8am to 4 pm, and 2 pm to 10 pm (ERROR: originally said 10AM) every day of the week (14 shifts). You may assume all nurses work 8 hours a day on a day (i.e., one shift) they are assigned.
- There should also be a Doctor assigned for 1 hour, every day of the week (basically, for writing prescriptions).
- Regulation also requires no more than 8 hours is assigned to a nurse in a single day. The checkCompliance() method of the CareHome class should throw an exception if these business rules are not enforced.
- Assume the same shift pattern will be repeated every week and a list of available qualified Doctors and Nurses are maintained in separate lists.
- Incorporate serialization to store objects created to file and to recreate them when the program is restarted.
- Write the JUnit tests for the relevant classes to ensure all the business rules are adhered to. Your test cases should include both positive and negative tests (where exceptions are expected to be thrown).
- Build a simple text-menu based system for the Aged Care facility manager to access and demonstrate all functionality you build at this stage. NOTE: that the text-based menu will be replaced by a GUI so you do not need to build both, unless you need to demo functionality BEFORE you build a GUI. (Any text-based menu build will be temporary and therefore does not need to be robust UNLESS you don't build a GUI interface.)

## 6. Use of Github

You are required to use Github repository management to manage your code.

- Create your repository via the Github Classroom link provided in Canvas Announcements on March 24 2021.
- You are expected to constantly update your code repository continuously, e.g., after each significant file modification (this may be several times per day, or even per hour).
- Easy-to-follow guides to using Github will be provided.

## 7. Submission format for final submission

- You will submit your assignment by uploading a zip file of your project via Canvas AND by providing a link to your private Github repository. Marks will be deducted if your Github repository shows development activity after the due date/time.
- You will also submit any related documents, e.g., a README detailing how to run your program, an outline of your design and design decisions made, and a brief refactoring report.

## 8. Academic integrity and plagiarism (standard warning)

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarized, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods,
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own. RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviors, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to the [University website](#).

## 9. Referencing guidelines

You are free to refer to textbooks or notes and discuss the design issues (and associated general solutions) with your fellow students or on Canvas; however, the assignment should be your OWN WORK.

The submitted assignment must be your own work. For more information, please visit

<http://www.rmit.edu.au/academicintegrity>.

Plagiarism is treated very seriously at RMIT. Plagiarism includes copying code directly from other students, internet or other resources without proper reference. Sometimes, students' study and work on assignments together and submit similar files which may be regarded as plagiarism. Please note that you should always create your own assignment work even if you have very similar ideas.

Plagiarism-detection tools will be used for all submissions. Penalties may be applied in cases of plagiarism.