

IFQ564 Data Structures and Algorithms

Assignment 1: Development of an inventory management system for a tool library

Word limit: 2000 words (+/- 10%) plus approximately 500 lines of code

Weighting: 60%

Due date: 11:59pm AEST Friday 23 September 2022 (Assignment Week)

After you have read this information, head over to the [Assignment Q&A](https://canvas.qutonline.edu.au/courses/1042/discussion_topics/75361) (https://canvas.qutonline.edu.au/courses/1042/discussion_topics/75361) discussion board to ask any questions and see what your peers are saying about this assignment.

Assignment overview

In this project, you need to develop a software application to solve a real-world problem. In the development of this software application, you need to use some basic data structures to store and manipulate data and need to apply relevant algorithms that are covered in this unit to solve those computational problems in the software application. Note that:

- The software application must be implemented in Microsoft Visual C#.
- You must not use any third-party C# class libraries.
- All the data structures used in the software application must be covered in this unit.

This assignment supports [unit learning outcomes 1, 2, 3 and 4](#).

Assignment details

You are hired to develop a software application for a not-for-profit tool library to manage the inventory of the tool library. All the tools in the tool library are bought through fundraising. Thus, the tool library rents its tools to local residents for free. In the tool library, there are nine different types of tools, including:

1. gardening tools
2. flooring tools
3. fencing tools
4. measuring tools
5. cleaning tools

6. painting tools
7. electronic tools
8. electricity tools
9. automotive tools.

Some tools are identical. Tools are considered to be identical if they have the same name. When the tool library purchases new tools, the staff member needs to add the new tools to the inventory management system. For a new tool, if there exists an identical tool in the tool library, the staff member only needs to increase the quantity of the tool in the inventory management system; otherwise, the staff member needs to enter into the inventory management system the following information:

1. the name of the tool
2. a brief description of the tool
3. the quantity of the tool.

When a person rents a tool from the tool library, the following details of the person need to be recorded in the inventory management system:

1. the full name
2. a contact phone number.

When the person returns the tool to the tool library, the record is removed from the inventory management system. In addition, when a person rents a tool from or returns a rented tool to, the tool library, the quantity of the tool in the inventory management system should be updated.

Functional and non-functional requirements

The inventory management system should meet the following functional and non-functional requirements:

- display the information about all the tools of a type selected by the user, including the name of the tool, a brief description of the tool, and the availability of the tool in alphabetical order by the tool names
- add new tools to the inventory management system
- lend a tool in the tool library to a person
- return a rented tool to the tool library
- a command-line user interface with a functionality menu.

Report structure

Your report needs to cover:

1. data structures used in your software application with justifications (explaining why the data structures are selected)
2. which algorithm you used, and how it is used, for implementing the function 'add new tools to the inventory management system'

3. your functional testing results including screenshots for each of the functional tests.

Submission details

Your submission should be a single zip file named your_student_number.zip, and comprising a complete C# console application project and your report.

Note: the reference list is not included in the word count.

Supporting resources

- [QUT cite|write: APA \(https://www.citewrite.qut.edu.au/cite/qutcite.jsp#apa\)](https://www.citewrite.qut.edu.au/cite/qutcite.jsp#apa).

Assignment criteria

1. data structures
2. algorithms
3. functionality
4. testing
5. code quality.

The marking guide outlines the criteria against which you will be assessed.

Assignment 1 marking guide

Assignment 1 marking guide

Criteria	7 High Distinction 85–100%	6 Distinction 75–84%	5 Credit 65–74%	4 Pass 50–64%	3 Marginal Fail 40–49%	2 Fail 25–39%	1 Low Fail <25%
Data structures (10%)	The data structures used in the software application are well chosen, well justified, and well presented.	The data structures used in the software application are well chosen, well justified, and reasonably presented.	The data structures used in the software application are well chosen, reasonably justified, and reasonably presented.	The data structures used in the software application are appropriately chosen, appropriately justified, and appropriately presented.	The data structures used in the software application are inappropriately chosen, with arguments. The presentation is reasonable.	The data structures used in the software application are inappropriately chosen with few arguments. The presentation is reasonable.	The data structures used in the software application are inappropriately chosen, without arguments. The presentation is poor.
Algorithms (5%)	The algorithm selection is correct. How the algorithm is applied is explained clearly and in detail.	The algorithm selection is correct. How the algorithm is applied is explained, but is lacking some detail.	The algorithm selection is correct. How the algorithm is applied is explained very briefly.	The algorithm selection is correct. How the algorithm is applied is explained inaccurately.	The algorithm selection is not quite correct.	The algorithm selection is incorrect.	No algorithm selection is included.
Functionality (65%)	All the required functionality has been implemented and all your implementations	All the required functionality has been implemented and most of your	Most of the required functionality has been implemented and the	Most of the required functionality has been implemented and the	All the required functionality has been attempted and the implementations meet the	Most of the required functionality has been attempted and the implementations	No required functionality has been attempted.

Criteria	7 High Distinction 85–100%	6 Distinction 75–84%	5 Credit 65–74%	4 Pass 50–64%	3 Marginal Fail 40–49%	2 Fail 25–39%	1 Low Fail <25%
	meet the requirements in the assignment specification. There is no runtime error.	implementations meet the requirements in the assignment specification. There is no runtime error.	implementations meet the requirements in the assignment specification. There is no runtime error.	implementations meet the requirements in the assignment specification. There are some runtime errors.	requirements in the assignment specification. There are compilation errors.	meet the requirements in the assignment specification. There are compilation errors.	
Testing (10%)	You have provided enough screenshots for every required functionality.	You have provided enough screenshots for most of the required functionality.	You have provided some screenshots for most of the required functionality.	You have provided some screenshots for some of the required functionality.	You have provided screenshots for a small number of the required functionality.	You have provided few screenshots for a small number of the required functionality.	No screenshot is provided.
Code quality (10%)	Your code is well structured and formatted, allowing the logic to be easily followed. Your code is clearly and concisely described by comments that fully document the code. Your code uses meaningful identifier names that enhance code readability by clearly explaining their purpose.	Your code is well structured, allowing the logic to be easily followed. Your code is clearly described by comments that fully document the code. Your code uses meaningful identifier names that enhance code readability.	Your code is formatted so that the logic can be followed with minimal effort. Your comments provide a good understanding of the code. Your code uses meaningful identifier names.	Your code is mostly formatted so that the logic can be followed with minimal effort. Your comments provide a general understanding of the code. Your code generally uses meaningful identifier names.	Your code is formatted so that it takes some effort to follow the logic. Your comments provide little understanding of the code. Your code uses too many unmeaningful identifier names.	Your code is not formatted. Few comments are provided. Your code uses too many unmeaningful identifier names.	Your code is not formatted. No comment is provided. Your code uses too many unmeaningful identifier names.

Assessment declaration

Assessment declaration and statement of authorship

In submitting this work I confirm that:

- It is my responsibility to check that I have submitted the correct assignment file before the due date.
- I am aware that any submission after the due date is considered to be late and is subject to the Late Assessment Policy. (See Syllabus for further details.)
- This work represents my individual effort and does not contain plagiarised material.
- I am aware that the University rule that a student must maintain academic integrity as stated and explained in [Academic integrity \(http://www.mopp.qut.edu.au/C/C_05_03.jsp\)](http://www.mopp.qut.edu.au/C/C_05_03.jsp).
- I am aware that my assignment may be stored in the Turnitin reference database, becoming part of the bank of material that assignments will be checked against in the future.

Submission details overview

This assignment will be submitted through Canvas. When you are ready to submit your assignment, select the 'Start Assignment' button at the top of this page. You will be taken to the 'File Upload' tab where you can choose your file or submit your URL.

If your assignment has multiple files, select 'Add Another File' for each subsequent file **before** ticking the 'Agree' checkbox and 'Submit Assignment'.

Please note: When you submit your assignment through Canvas, you are also submitting the assignment through Turnitin, which is a text-matching service that compares your work with an international database of information sources. You will need to agree to using it.

Assignment support

Don't forget that in addition to your OLAs who provide discipline-specific content advice, you can access the 24/7 draft writing service from Studiosity.

If you need assistance with academic feedback on a draft of your assignment, see [Assignment support: Studiosity](#).

Request Assignment Extension

We understand that at times your studies may be adversely affected by illness, misadventure or some other extraordinary cause or circumstance reasonably beyond your control. If you feel this is the case, and you need an extension to support you in completing your assignment, please select "I would like to request an extension" below to access the request process.

Please note:

- For team assignments, each team member must submit a request.

[I would like to request/view an extension \(../external_tools/1101\)](#)

► [Request Assignment Extension](#)