Assignment Q & A

This document contains questions and answers regarding the assignment, and will be updated with new entries as and when they arise. Note, however, that questions for which the answers may be found in the Assignment Specification, Assignment Report Template and/or Assignment Rubrics will not need to be addressed here. In addition, to avoid redundant work, we will not be responding to questions asked in emails/Classroom/Google chat if the information may be obtained from the Assignment Specification, Assignment Report Template and/or Assignment Rubrics.

This Q&A has been grouped by topic which you can conveniently access via the document outline on the left of your screen:

| First, click on “Show document outline”: | Next, click on the link for the relevant topic: |
| --- | --- |

Links to newly added Q&A entries (in LIFO order):

* [QUESTION: Is it okay if a team member does not use our team collection ADT?](#_8iv7w2djg7b0)
* [QUESTION: In my subsystem, may I use the same collection ADT that my team member used in his subsystem?](#_6kcqqkha12ev)
* [QUESTION: In my subsystem, may I use more than 1 collection ADT?](#_7c4tcgn2o8pq)
* [QUESTION: May we use the collection ADTs provided in the sample code in our subsystems?](#_8ornlu4qj540)

# Collection ADTs

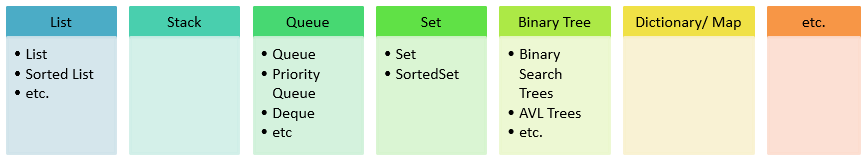
## General Questions

### Purpose

What is the point of implementing collection ADTs by ourselves when the code is simply given in the sample code: what is preventing us from just copying whatever java interface or class we need for our ADT? Is that considered plagiarism?

ANSWER:

* The point is to learn about collection ADTs (what are the different collection ADTs, what are the different ways of implementing collection ADTs, etc.).
* If you use the collection ADT implementation from the sample code or from other sources, do acknowledge that at the top of your Java interfaces and/or classes.
* The code given in the sample code is just one example of many ways to implement some of the basic collection ADTs. There are many alternative ways of implementing such collection ADTs. In addition, there are many other collection ADTs - refer to the following non-exhaustive list for some examples:



* You may even create your own collection ADTs with the required characteristics and operations.
* If students choose to directly use the collection ADT implementation given in the sample code, their marks for assessment criteria “Collection ADT implementation - originality” would be considered to be in the “Developing” category. Refer to the Assignment Rubrics for details.

### What does i*mplementation approach* mean?

“Implementation approaches” refers to the different ways of implementing the collection ADTs. For example, here is a non-exhaustive list of different implementation approaches for the List ADT:

* Linear array implementation
* Circular array implementation
* Linear linked implementation
* Doubly linked implementation
* Circular linked implementation
* Circular doubly linked implementation
* Linked implementation with dummy/header node

### Can we use java collection frameworks such as

java.util.List,ArrayList,Arrays,Stack,Queue,Deque, Vector, priorityqueue,Linkedlist, Linkedstack,Hashmap,Dictionary, functions, functional, Hashset, predicate,others collection etc. in our assignment?

ANSWER: No, because the assignment requires you to learn about creating and implementing user-defined collection ADTs. Therefore, you may not use any predefined collection interfaces and classes from the Java Collections Framework.

### Is it OK to use **Comparator** and **Collection.sort()** to do sorting or we must do our own sorting algorithm like quicksort and mergesort?

ANSWER:

**Comparator** - Yes.

**Collection.sort()** - No.

If you need to use a sorting operation, then you should implement your own.

### Can we use Java’s predefined classes and interfaces such as **String**, **Double**, **Character** and **GregorianCalendar**?

ANSWER: Yes, you may use any Java interfaces and classes that are not collections.

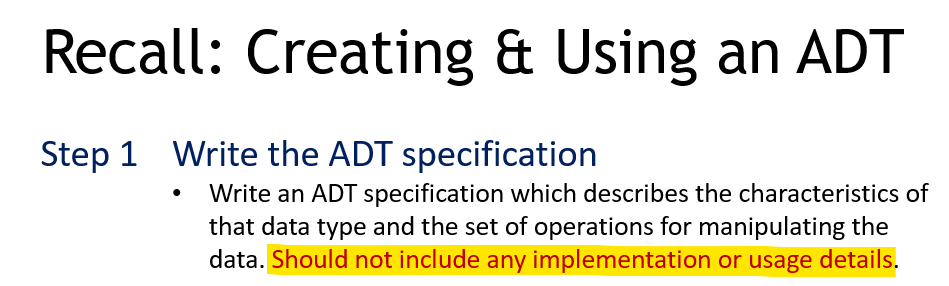
### Can we use **java.util.Iterator**?

ANSWER: Yes, you may use any Java interfaces and classes that are not collections.

~ This section ends here. The next section starts on a new page. ~

# ADT Specifications

## Good and Bad Practices



| Good and Bad Practices  *Note: Bad practices are highlighted in yellow.* | |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  | addCustomer(int id, String name, String mobile)  Description: Add a new customer to the list.  . . . |
|  | ADT *CustomerList*  This list is to manage a list of customers for a online store.  . . . |

## Operations to include

### Need to include basic operations I don’t use?

QUESTION: If there are certain basic operations of the collection ADT that I don't use in my control or entity classes, do I need to include them in the collection ADT’s specification, Java interface and Java class?

ANSWER:

* YES. Your ADT should contain all the basic operations for that ADT for completeness. Remember the principle of reusability ... your ADT may be used in future projects for different applications, different entity objects, etc.
* However, although it is not compulsory to use all the basic collection ADT operations in your control and/or entity classes, it would be good if you can use your creativity to demonstrate meaningful use of the ADT operations where possible in your control and/or entity classes.

### 

### Need to include the iterator operations in the ADT specification?

QUESTION: Do we need to include the iterator operations (i.e., hasNext(), getNext()) into the ADT Specification if we implemented the iterator for our collection ADT?

ANSWER: Do not include any operations that do not belong to the collection object. The hasNext() and getNext() operations belong to the Iterator, not the collection.

### 

### Do we need to include the getIterator() operation in the ADT specification?

QUESTION: Do we need to include the getIterator() operation in the ADT specification?

ANSWER: Yes, since it is meant to be used by the clients of the collection.

~ This section ends here. The next section starts on a new page. ~

# 

# ADT Implementation

## Java Interface

| Good and Bad Practices  *Note: Bad practices are highlighted in yellow.* | |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

#### 

#### 

## 

## Java Class

### Good and Bad Practices

| Good and Bad Practices  *Note: Bad practices are highlighted in yellow.* | |
| --- | --- |
|  |  |
|  |  |

#### 

#### 

### Sorting , Searching algorithms

QUESTION: Will I get additional marks if I use a sorting, searching algorithm ?

ANSWER: Depends on the appropriateness of implementation and use. Marks will be awarded according to the assessment criteria and corresponding descriptors specified in the Assignment Rubrics.

~ This section ends here. The next section starts on a new page. ~

# 

# Modules(Entity classes, Control classes)

## Use of One Collection ADT for team

1. QUESTION: Will I get more marks if I use more than one collection ADT in my subsystem?

ANSWER: Use ONE is fine. You should use a team’s collection ADT with add on methods. Marks will be awarded according to the assessment criteria and corresponding descriptors specified in the Assignment Rubrics. If your use of multiple collection ADTs fulfills the ‘Ideal’ descriptors of any assessment criteria, then you will be awarded marks in the ‘Ideal’ range for the relevant assessment criteria. A word of caution: please read and understand the descriptors in each assessment criterion specified in the Assignment Rubrics and avoid having a one-dimensional assumption that “If I do …, I will get higher marks in the assignment”. This is as we’ve observed students who focus only on one aspect of the assignment and ignore other parts, resulting in the students receiving lower overall marks. Therefore, we would like to reiterate and stress that **marks will be awarded according to the assessment criteria and corresponding descriptors specified in the Assignment Rubric**s.

### May we use the collection ADTs provided in the sample code in our subsystems?

ANSWER: Yes.

### How many collections of ADT I use in my subsystem?

Use ONE Collection of ADT that is selected by Team.

Each member should contribute to the implementation of ADT methods (modified, add new) in the Implementation class and invoke the methods from the collection ADT to complete your module.

### Is it okay if a team member does not use our team collection ADT?

Not encouraged, all members should use the team's 1 collection ADT, including the implementation and also how to use the ADT.

## Entity Classes

1. QUESTION: Should constructors, setters, getters, toString, and equals methods be included in each entity class?

ANSWER: Yes,it is a good OO programming practice to include constructors, setters, and getters in each entity class. In addition, you should also override the toString, and equals methods in each entity class. Some methods will not work correctly if you did not override the equals and compareTo methods.

1. QUESTION: Can entity classes include input statements (to obtain input from the user) and/or output statements (to display output to the user)?

ANSWER: NO. This is bad design. Entity classes should NOT include any input statements to obtain input from the user and should NOT include any output statements to display output to the user. The best practice is to make each entity class a POJO (plain old Java object) so that they can be independent of the entity classes and control classes that use them.

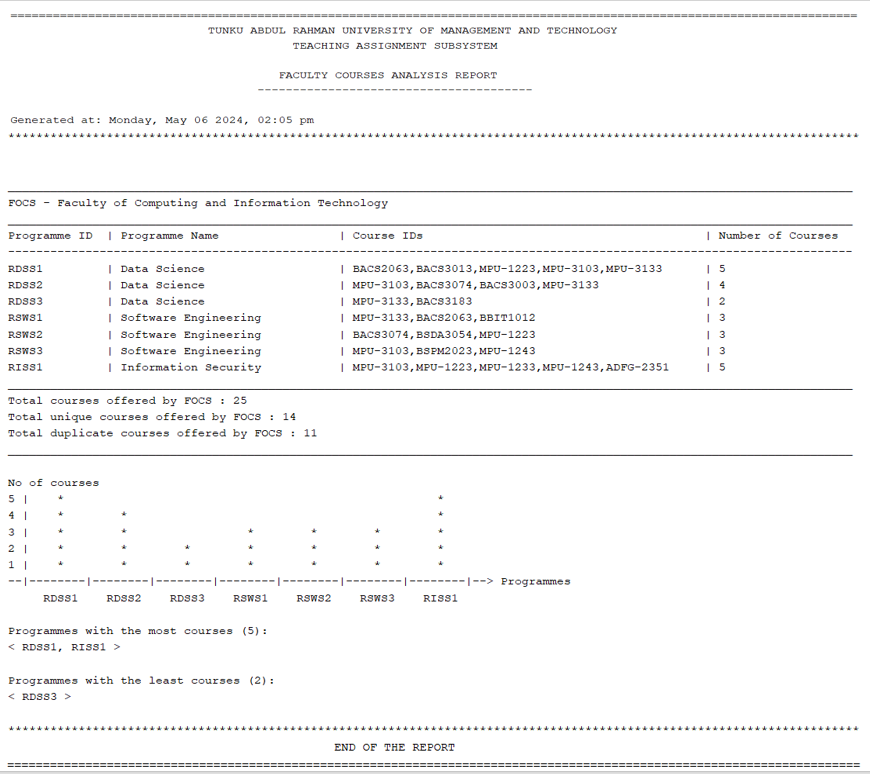
An entity class represents the template for objects that you will store to persistent storage (e.g. binary files, database table, etc.). Examples of entity classes are Runner, Customer, etc. Entity classes SHOULD NOT include statements to obtain input from the user (e.g. using Scanner) or displaying output to the user (e.g. using System.out.println).

## Database / Files

1. QUESTION: For the database, do we need to use the built-in Apache database or do we need to specially download the database software to connect to databases such as SQL Server?

ANSWER: You are not required to use a database for this assignment. If you do, you may use any open-source database.

## Report feature



## Miscellaneous questions

1. QUESTION: Do we need to do type validation for the user input?

ANSWER: You are required to include only validations which involve the invocation of a method of the collection object.

1. QUESTION: For the Generate Report functionality, do I have to use other software?

ANSWER: No, you are not supposed to use other software. Instead, students are required to write code in Java to generate the reports.

1. QUESTION: For “**Generate summary reports**”, can I just list all the entries in the collection object?

ANSWER: No, because listing all entries in the collection object is already specified as a separate use case under the subsystem. If you want to get Ideal or Approaching marks for the criteria “*Use of ADTs - creativity and complexity*”, all the implementations of your use cases should be such that it involves creativity and complexity that demonstrates a high level of understanding and competency in using collection ADTs.

1. QUESTION: Why is it better to use the Java interface instead of the implementation class when declaring collection variables or parameters (see below for example)?

**ListInterface**<Employee> employeeList;

ANSWER: It's best practice to use the Java interface (e.g. **ListInterface**) as the type for the variable because

1. this declaration restricts the allowed methods on the variable to only the methods declared in **ListInterface** and
2. enforces the requirement for the implementation classes (e.g., **ArrayList**, **LinkedList**, etc) to provide the implementations of all the abstract methods in the interface.

Moreover, if we used the interface instead of the class as the type for method parameters, e.g.

void displayList(**ListInterface**<Employee> employeeList)

we could replace the class (e.g., ArrayList, LinkedList, etc) with another class that also implements ListInterface without modifying the subsequent statements in the program.

~ This section ends here. The next section starts on a new page. ~

# Assignment Report

1. QUESTION: For the complete source code (for collection ADT implementation, control classes, entity classes), can I embed screenshots of the source code in my report?

ANSWER: Please refer to your tutor’s requirements.

General answer:

Yes, as long as they can be easily and clearly read. Do ensure that:

* Use light mode instead of dark mode in your IDE.
* Ensure that there is proper indentation for your class. To automatically format, either
  + Right-click on your mouse and then select **Format**, or
  + Press <Alt><Shift><F>
* The font size is big enough (or you can zoom-in by pressing the Alt key and then using the scroll wheel on your mouse) before you capture the screenshot using either the *Snipping Tool* or *Snip & Sketch* tool that is automatically provided by Windows.

1. QUESTION: For the documentation, do we need to write the overview of the chosen application or the specific subsystem that we worked on?

ANSWER: NO.

1. QUESTION: Is there any guideline for the documentation format like: Times New Roman, 12 point font size and so on?

ANSWER: Use the font type and size in the Assignment Report Template.