FACULTY OF ENGINEERING AND INFORMATION SCIENCES

SUBJECT'S INFORMATION:			
Subject:	CSIT121 Object Oriented Design and Programming		
Session:	Autumn 2019 (February)		
Programme / Section:	Computer Science and IT		
Lecturer:	Ms. Siti Hawa		
Coursework Type *(tick appropriate box)*	❑ Individual Assignment ❑ Group Assignment ❑ Project ✓ Lab Task ❑ Seminar / Tutorial Paper ❑ Others		
Coursework Title:	**Lab Task 3**	Coursework Percentage:	3%

ASSESSMENT CRITERIA:	
Correctness	All programs should produce the correct result as stated in the specification.
Coding	Programs should be written only using sequencing structures involving declarations, input/output statements, and assignment statements.
Readability	Appropriate comments are included. Meaningful identifiers used. Proper indentation and line spacing used.
Well formatted output	Output should be well formatted with appropriate messages displayed. Numbers are shown with appropriate precision.

SUBMISSION:

All completed work should be submitted online through Moodle before or on the due date provided.

SUBMIT AS EARLY AS POSSIBLE. ONLY ONE SUBMISSION IS ALLOWED. IF RE-SUBMISSION IS NECESSARY, YOU ARE REQUIRED TO REMOVE THE EARLIER SUBMISSION AND THIS MUST BE DONE BEFORE THE DUE DATE. OTHERWISE YOU WILL BE PENALIZED FOR LATE SUBMISSON.

DUE DATE:	**WEEK 8**

PENALTIES FOR LATE SUBMISSION:

Penalties apply to all late work, except if student academic consideration has been granted. Late submissions will attract a penalty of 25% of the assessment mark per day including the weekend. Work more than (3) days late will be awarded a mark of zero.

PLAGIARISM:

When you submit an assessment task, you are declaring the following

1. It is your own work and you did not collaborate with or copy from others.
2. You have read and understand your responsibilities under the University of Wollongong's policy on plagiarism.
3. You have not plagiarised from published work (including the internet). Where you have used the work from others, you have referenced it in the text and provided a reference list at the end ot the assignment.

COURSEWORK SPECIFICATION

OBJECTIVES:

Following completion of this task, students should be able to:

- Design and implement abstract class and interface

Question 1

Declare and implement an **abstract class** called `Ship` that has the following:
- An attribute to represent the `name` of the ship.
- An attribute to represent the `year` that the ship was built.
- A default constructor to initialize the attributes.
- Accessor methods and mutator methods for each of the attribute.
- An **abstract method** called `displayDetails()`.

Declare and design a subclass of `Ship` called `CruiseShip`. The `CruiseShip` class should have the following:
- An additional attribute to keep the maximum number of passengers.
- A default constructor to initialize the attribute in `CruiseShip` as well as the `Ship` class.
- Accessor methods and mutator methods for its attribute.
- The `displayDetails()` method that will display all details from `Ship` and `CruiseShip`.

Declare and implement another subclass of Ship called `CargoShip`. The `CargoShip` class should have the following:
- An attribute to keep the cargo capacity in tonnage.
- A default constructor to initialize the attribute in `CargoShip` as well as the `Ship` class.
- Accessor methods and mutator methods for its attribute.
- The `displayDetails()` method that will display all details from `Ship` and `CargoShip`.

Write a `main()` method in another class called `ShipApp`. Declare an `ArrayList<Ship>` to keep several `CruiseShip` and `CargoShip` objects. Use a simple menu to allow the user to

(1) add either a `CruiseShip` or a `CargoShip`
(2) edit either the number of passengers for a `CruiseShip` or the cargo capacity for the `CargoShip`
(3) display all the objects in the `ArrayList`.
