

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 <small>(tick appropriate box)</small>	<input checked="" type="checkbox"/> Individual Assignment	<input type="checkbox"/> Group Assignment	<input type="checkbox"/> Project
	✓ Lab Task	<input type="checkbox"/> Seminar / Tutorial Paper	<input type="checkbox"/> Others
Coursework Title:	Lab Task 2	Coursework Percentage:	3%
ASSESSMENT CRITERIA:			
<p>All programs should produce the correct result as stated in the specification. Programs should be written using appropriate control structures, and using correct class and inheritance implemented. Meaningful identifiers used. Proper indentation and line spacing used. Suitable comments are recommended. Output should be well formatted with appropriate messages displayed. Numbers are shown with appropriate precision. Programs with syntax error and are unable to execute will not be awarded any mark.</p>			
SUBMISSION:			
<p>All completed work should be submitted online through Moodle at the end of your lab class. One day extension may be granted upon request. Extended work will not be eligible for full mark. Failing to submit within the specified time will result in zero (0) mark.</p> <p>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 SUBMISSION.</p>			
DUE DATE:	WEEK 6		
PENALTIES FOR LATE SUBMISSION:			
<p>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.</p>			
PLAGIARISM:			
<p>When you submit an assessment task, you are declaring the following</p> <ol style="list-style-type: none"> 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 of the assignment. <p>Plagiarism will not be tolerated. Students are responsible for submitting original work for assessment, without plagiarising or cheating, abiding by the University's policies on Plagiarism as set out in the University Handbook under University Policy Directory and in Faculty handbooks and subject guides.</p>			

COURSEWORK SPECIFICATION

OBJECTIVES:

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

- Design and implement classes using inheritance
-

Question 1

Design and implement a class called `Vehicle` that has a registration number, manufacturer's name (type `String`), and the engine (type `String`). Provide a default constructor to initialize all attributes as well as the set method for each attributes. Implement also a method to display the `Vehicle` details.

Next, create two subclasses that inherit from `Vehicle`. The first subclass is called `Truck`. A `Truck` class has additional properties that includes the load capacity in tons (type `double`) and a towing capacity in tons (type `double`). Provide a default constructor for this class that will initialize its attributes as well as the superclass attributes. Provide also the set methods for its attributes and another display method to overload/override the one defined in the superclass so that it will also display the `Truck`'s attributes as well as the `Vehicle`'s.

The second subclass is called `Aircraft`. An `Aircraft` object should have as additional attributes the type of aircraft (cargo, passenger, etc), and the capacity (num of passengers, baggage, cargo, or fuel). Also provide a default constructor, set methods, and display methods similar to the `Truck`'s class.

Write another class that contain a `main()` method to test your classes. Your `main()` method should create an `ArrayList<Vehicle>` as a general type. Create several `Truck` and `Aircraft` objects and store them in the `ArrayList`. Use the set methods to change the attributes for each objects and display all object's details on the screen.
