

FACULTY OF ENGINEERING AND INFORMATION SCIENCES

SUBJECT'S INFORMATION:			
Subject:	CSCI251 Advanced Programming		
Session:	July 2019		
Programme / Section:	Computer Science		
Lecturer:	Ms. Siti Hawa		
Coursework Type <small>(tick appropriate box)</small>	<input type="checkbox"/> Individual Assignment <input checked="" type="checkbox"/> Lab Task	<input type="checkbox"/> Group Assignment <input type="checkbox"/> Seminar / Tutorial Paper	<input type="checkbox"/> Project <input type="checkbox"/> Others
Coursework Title:	Lab Task 8	Coursework Percentage:	1%
ASSESSMENT CRITERIA:			
<p>All programs should produce the correct result as stated in the specification. Programs should be written only using the programming structures and concepts already covered during lectures. Meaningful identifiers used. Proper indentation and line spacing. 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 before the due date provided.</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 13		
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:

- Write C++ programs using polymorphism and abstract classes.
-

Question 1 (Polymorphism and abstract classes)

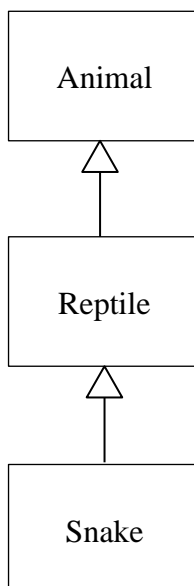
Declare and implement an abstract class called `Car`. The `Car` class should have the car registration number and the number of miles the car has traveled. Include in the class a function to set the car registration number, a function to increase the miles traveled by receiving the number of miles traveled in the last trip, an accessor function to return the car registration number, another accessor function to return the miles traveled, and a **display function** that will be implemented by the **derived** classes of `Car`.

Declare and implement a derived class from `Car` called `RaceCar` that has an additional data member called `racingType` that may have values such as track racing, off-road racing, drifting, or kart-racing. Include a constructor to initialize its data member as well as to set the derived data members from the `Car` class. Implement a display function for the `RaceCar` class to display all its details.

Declare and implement a second derived class from `Car` called `MPV` that has an additional data member called `capacity` that represents the number of people that can fit into the car. Include also a constructor to initialize its data member as well as to set the derived data members from the `Car` class. Implement a display function for the `MPV` class to display all its details.

Question 2 (Polymorphism and abstract classes)

Define three classes with the following relationships:



Animal should be abstract. It should hold two data members, name and size to keep the weight of the animal. Reptile should have habitat as data member to keep information of the habitat such as land, salt water, fresh water, etc Lastly, Snake should be a class with poisonous as data member to indicate whether it is poisonous or not.

There should be a showInfo() function which is overridden through the hierarchy. It should provide appropriate information about the objects. Classes Reptile and Snake should have a non-default constructor to initialize the data members, while Animal needs a function to set its data member.

The main() function should construct 2 instances each of Reptiles and Snake. These should be stored in an array of type Animal. A loop should be used to output information on all the objects, producing something like:

```
-----  
Crocs weighs 225 kg and lives in salt water  
-----  
Lizards weighs 0.005 kg and lives on land  
-----  
Cobra weighs 9 kg and lives on land. It is very poisonous.  
-----  
Anaconda weighs 300 kg and lives in swamps  
-----
```
