FACULTY OF ENGINEERING AND INFORMATION SCIENCES

SUBJECT'S INFORMATION:			
Subject:	CSCI251 Advanced Programming		
Session:	Spring 2019 (July)		
Programme / Section:	Computer Science		
Lecturer:	Ms. Siti Hawa		
Coursework Type *(tick appropriate box)*	❑ Individual Assignment ❑ Group Assignment ❑ Project ✓ Lab Task ❑ Seminar / Tutorial Paper ❑ Others		
Coursework Title:	**Lab Task 5**	Coursework Percentage:	1%

ASSESSMENT CRITERIA:

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.

SUBMISSION:

All completed work should be submitted online through Moodle before 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 10**

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.

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.

COURSEWORK SPECIFICATION

OBJECTIVES:

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

- Write C++ programs that involves writing class and objects.
- Write C++ programs using class associations.

Question 1 (Class and Objects)

Write a C++ program to implement a class named `Troll`. The `Troll` class should have the following properties:

- Data fields for name, type of Troll, and age.
- A private static field for the Troll toll, which has an initial value of RM15.00.
- A default constructor.
- A static function to change the Troll toll that takes a new toll value as parameter.
- A `setField()` function to change the name, type, and age of a Troll.
- A display function to display a Trolls details and the Troll toll.

Write a main function that creates several Troll objects, set and display the information. Then change the toll value and display all the Trolls information again. A sample output may look like the following.

```
Boris is a Mountain troll!!
Boris is 8 years old.
The Troll toll is RM15.00.

Igor is a Forest troll!!
Igor is 100 years old.
The Troll toll is RM15.00.

Enter new value for the toll: RM30.00

Boris is a Mountain troll!!
Boris is 8 years old.
The Troll toll is RM30.00.

Igor is a Forest troll!!
Igor is 100 years old.
The Troll toll is RM30.00.
```

Question 2 (Class Associations)

For this question, you are going to write three classes and relate them through associations.

First, create a `Person` class that includes fields for first name, last name, and age. Include a non-default constructor and a display function.

Second, create a `Date` class that contains three integer data members to represent the month, day, and year. Include also a non-default constructor and a display function for this class.

Lastly, create a class called `DentalAppointment`. This class should have fields for patient (an object of `Person` class above), date of the appointment (an object of the `Date` class above), and the duration of the appointment in minutes. The `DentalAppointment` class should also have a non-default constructor that receives the details of a `Person`, the details of a `Date`, and the duration as parameters. It should also have a display function to display the details of an appointment.

Write a `main()` function that creates a `DentalAppointment` object and display the details of the appointment.
