

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 <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 7	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 12		
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 inheritance.
-

Question 1 (Inheritance)

Declare and implement a class named `Payment` that contains a member variable of type `float` that stores the amount of the `payment`, a default constructor, a non-default constructor, a `setField()` function that receives parameters to set the data member, and a member function named `paymentDetails()` that outputs the amount of the payment.

Next, declare and implement a class named `CashPayment` that is derived from `Payment`. Include `balance due` as the data member for this class. This class should override the `paymentDetails()` function to add a message to indicate that the payment is in cash and display the balance due if any. Include also a default constructor, a non-default constructor and overload the `setFields()` function.

Declare and implement another class named `CreditCardPayment` that is also derived from `Payment`. This class should contain member variables for the name on the card, expiration date, and credit card number. Include a default constructor, a non-default constructor, overload the `setFields()` functions, and finally, override the `paymentDetails()` function to include all credit card information in the printout apart from the payment amount.

Create a main function that creates two `CashPayment` and two `CreditCardPayment` objects with different values. Show that some objects are initialized using the non-default constructor and some are initialized using the default constructor and then set using the `setFields()` functions. Display the payment details for each of the objects.

Question 2 (Multiple Inheritance)

Write a C++ program to implement multiple inheritance. There should be two base classes: `House` and `Investment`.

The `Investment` class should contain fields to hold the initial value of an investment, the current value, the profit (different between the previous two fields), and the percentage profit (profit divided by initial value). It should have a non-default constructor that requires parameters for the initial and current values, and a display function.

The `House` class should have fields for street address and size (in square meters). It should also have a non-default constructor that sets both fields using values passed as parameters, and a display function.

Create also a `HouseInvest` class which inherits from both `Investment` and `House`. It includes a non-default constructor that receives arguments for a house details and also an investment details, and a display function which uses both the base classes display functions.

Write a `main()` function that declares a `HouseInvest` object and displays its values.
