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 6**	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 11**

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 using friends and overloaded operators.

Question 1 (Friends)

Write a C++ program with two classes.

1. The first, `Customer`, holds customer data, specifically an identification number and name.
2. The second, `City`, holds city information, specifically the city name, state, and postal code.
3. Each class should have a constructor that takes arguments to set the field values.
4. Create a `friend` function that displays a customer identification number, name, city, state, and postal code.

Write a `main()` function that creates a `Customer` object and a `City` object. Then call the display function to display the details of both objects.

Question 2 (Overloaded Operators)

For this question, you are required to create a class called `CollegeDepartment`. The class should have fields for the department name (for example, "Informatics"), The department dean's name (For example, "Jane Lim"), an integer that holds the number of courses offered by the department (for example, 5), an integer that holds the total number of students, and a string pointer that dynamically creates a list of course names depending on the number of courses stated earlier. An example of a course name is "CSCI251".

Include a default constructor, an overloaded insertion operator (<<), and an overloaded extraction operator (>>) that prompts the user for the department name, dean's name, the number of courses, and after dynamically creating the string list, prompt the course names as well.

Overload the operator + that will add an integer to the total number of students for the department. The operator should be used as follows:

```
CollegeDepartment dept1;
dept1 = dept1 + 200;  //200 will be added to the total number of students
```

Overload also the > operator that determines which `CollegeDepartment` object has more courses offered. Include also an overloaded = operator to assign one `CollegeDepartment` to another.

Write a suitable `main()` function to test all the functions and operators for the class.
