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 9**	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 15**

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 exceptions and function templates.

Question 1 (Exception Class)

Create an exception class called InventoryException that inherits from runtime_error. The class should contain three constructors, a default constructor and two non-default constructors. The default constructor will pass the message "Invalid stock number. Stock number should be between 1111 and 9999" to the what() function in the base class. One non-default constructor should except an integer value that will pass the message "Invalid quantity. Quantity should not be negative" to the what() function. And another non-default constructor that accepts a floating-point value that will pass the message "Invalid price. Price should not be below RM0 and over RM10000.00" to the what() function.

Create an Inventory class with data members for stock number, quantity, and price. Include a default constructor, and overloaded insertion and extraction operator. The extraction operator should throw a suitable InventoryException object if the stock id, quantity, and price entered are invalid.

Create a main() function that instantiate an Inventory object, read its data and display the data of the object if no exception is thrown. Perform exception handling method in main().

Question 2 (Function Template)

Write a template version of the iterative binary search algorithm that searches an array of arbitrary type for a given key.

Declare and implement a class called Student that keeps the student id, name, and grade. Include a default constructor, the overloaded insertion (<<) operator and also the overloaded extraction operator (>>).

Declare and implement another class called Book that keeps the book's title, author, and price. Just like the Student class, Include in class Book a default constructor, the overloaded insertion (<<) operator and also the overloaded extraction operator (>>).

Write a main() function that test your function by passing parameters of type int, float, char, Student, and Book. Identify and implement additional suitable operators needed to be overloaded in the Student and Book class that you define above so that it can work well with the template binary search function that you have written. A Student object should be searched using the student id and the Book object should be searched using its title.
