FACULTY OF ENGINEERING AND INFORMATION SCIENCES

SUBJECT'S INFORMATION:			
Subject:	CSCI251 Advanced Programming		
Session:	Spring 2019 (July Session)		
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:	**Assignment 2**	Coursework Percentage:	10%

ASSESSMENT CRITERIA:		
Correctness	All programs should produce the correct result as stated in the specification.	2 marks
Class Design and Inheritance	Class declared and implemented as specified complete with proper validation on data. Appropriate data hiding and encapsulation implemented. Correct Inheritance implemented.	2 marks
Polymorphism	Polymorphism implemented with suitable virtual functions.	2 marks
Overloaded Operators and Friends	At least 3 suitable overloaded operators implemented correctly and at least one friend function is used.	1 mark
Main Function	Good design of the main function with complete functionalities included. Well-structured and used modular approached. Necessary data validation is implemented.	2 marks
Readability and Well Formatted Output	Appropriate comments are included. Meaningful identifiers used. Proper indentation and line spacing used. Output shown is easy to understand provide sufficient information.	1 mark

SUBMISSION:

All completed work should be submitted online through Moodle before or on 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:	**Friday, 25th October 2019 (11:55 pm)**

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. under University Policy Directory and in Faculty handbooks and subject guides.

COURSEWORK SPECIFICATION

OBJECTIVES

This assignment aims to provide you with some experience in writing codes using C++ programming language that covers the following topics:

- Inheritance and Polymorphism
- Overloaded operators and friends

Remember that:

1. **All programs should be able to run on the lab's computers.**
2. **You must put the following information on the header of each text and source file you will be submitting in this assignment:**
   Student's full name:
   Student's ID:
   Modification Date:
   Purpose of this file (or program):
3. **Assignments that are not able to be compiled will result in zero mark given to the assignment.**
4. **You must only use the C++ features that have already been covered in the lectures**

Question:

For this assignment, you are required to write C++ codes to represent a simple application for a fantasy role-playing game. In this game, you have four different type of creatures: wizard, elf, dwarf, and demon. Each creature has attributes to represent its name, strength and hit points. The strength is used to measure how much damage this creature can inflict on others while the hit points is to represent how much damage this creature can sustain.

In a particular game, when a creature attacks, we can request to get the damage value the creature can inflict. Your program should decide on the general strength and hit point value when a creature is created and this will be the initial value for all creatures.

The damage value is determined by its strength. However, the damage can be more depending on the type of the creature. Demons can inflict 50 additional damage with a 5% chance. Dwarf inflict double damage with a 10% chance while Elf are fast enough that they get to attack twice. Wizard can inflict magical damage that causes their opponent to miss their next attack with either 50% chance if no wand is used or 90% chance when a wand is being used.

Every time a creature is attacked, the hit points are reduced based on the damage caused by the creature attacking it. If the hit points reaches 0 or less, the creature is pronounced dead and the object representing the creature should be destroyed. There can be more than one type of creature in a game played.

Your application should allow several games played one after another. In each game, the user may select any two creatures to be created and and fight. The winner may stand a chance to fight another chosen opponent or the user may choose to quit the game.

The following are the requirements that must be implemented in the application

- Design and implement classes using inheritance to represent the creatures. The base class should be made abstract. Identify the suitable attributes for each classes. All derived classes must have unique attributes apart from the attributes inherited from the base class. Identify and implement suitable constructors, accessors, mutators, and also abstract method(s).
- You must also include suitable overloaded operators to your classes. At least three different overloaded operators should be used.
- There must be at least one friend function declared.
- Polymorphism must be implemented. To do this, all the creatures created must be stored in a single array. There must be demonstration on the use of polymorphic call and also dynamic cast in your application.

You are free to demonstrate how the game can be played. However, the following should be present in some ways:

- Status of each creature (or at least the creatures involved in the current fight)
- The attacking creature and the creature being attacked
- The damage value when an attack is made
- The winning creature
- The creature that is destroyed

Submission requirements:

(i) Design your classes and draw a simple UML class diagram to show all the classes involved and their relationships. Include brief explanation on your design. You are required to show your diagram during class on Tuesday 15/10/2019.

(ii) Implement you application as specified above. Your codes should be placed in separate header and implementation files. The main() function should be placed in a file named main.cpp.

(iii) You are required to explain your class design and demo the game played during the demo session.