**Data Structures (2080C) – Lab 8**

***Topics covered: Templates and Class***

**Objective:**

The objective of this Lab is to create an implementation of a D&D like fighting game.

**Requirements:**

1. **All classes that have data members should have at least an accessor to query the data, and potentially a setter if it makes sense.**
2. You will need to create classes for the following concepts.
   1. An Interface that represents Race – Elves, Dwarves, Humans, Halflings
      1. These will have three methods on them HitPointModification, ArmorClassModification, ,HitBonusModification. Initiative Bonus
   2. An Interface class that represents a Character, this will inherit from Race Interface
      1. They have Hit points, armor class, to hit bonus, and initiative bonus
      2. They will have accessors to retrieve Hit points, Armor Class, Hit Bonus, that will take into account Racial Modifications
      3. Will have (most likely pure virtual methods) to Attack, take damage,
   3. Concrete classes that inherit from characters to represent Wizards, Rogues, Fighters, Clerics that represents (yes if you have D&D background, these are called classes 😊)
      1. Make some behavior changes for these character classes. Ex. Fighters have more HP and have higher AC. Rogues deal more damage and can hit back row of opposing party, wizards hit all opposing party members but low damage, and clerics can heal. Feel free to add more exceptions, this is just a starter list. I want you to have some design reign here 😊
3. Create a Template to an adventuring Party

And adventuring party consists of four characters with two in the front row, and two in the back row. Should be able to make a Party of Characters, Elves, only Wizards, and etc. As you develop the attack system below, feel free to add method to the Template make things easier (i.e. damage entire party, pick a party member in rear, or front)

1. Make a executable that will allow one to setup to two opposing parties, and have them fight. Some rules we will have from D&D :
   1. Default AC is 10
   2. To attack someone, we pick a random number 1-20 and add to hit bonus. And then compare against targets AC.
   3. Unless they have a special ability, character attacks opposing parties front line characters until they die/knock out, and then works backwards. Make it easy, and say front row must be incapacitated before hitting anyone in back row
   4. Have some basic AI so attacking is automated (so we can answer the question can a party of wizards beat up a party of fighers)

The requrements is not compete, as I did this on purpose. You will need to make assumptions or decision points you had to make to implement this. These will need to be documented in the lab report.

**Lab Submission:**

1. Package all files in a single zip folder and submit the file as a group via Blackboard.
2. **Lab report contains** 
   1. **class diagrams (use doxygen to generate easily)**
   2. **Assumptions or changes made to requirements above. Try not to go overboard here, bulleted list of assumptions\phrases is OK. No need for a 10 page requirement document 😊**
3. **If you were going to add in a new race and character class, what would it be? What abilities\exceptions\weaknesses would it have? Would your existing framework support it or what modifications would you need to make.**

**Lab Grading:**

1. 20% - Lab attendance
2. 60% - Task 1 has been correctly implemented and meets all requirements.
3. 20% - Lab report contains all required information and is well written.

If program fails to compile, 0% will be given for that Task.