Data Structures (2028C) **Lab 4 -**

Fall 2020 (Week 5, 9/22-24) ***Inheritance, Polymorphism and Abstract classes***

**Objective:**

The objective of this Lab is to examine generalization/specialization in OOA/D/P: deriving a new class from an old one through inheritance, polymorphism, and abstract classes.

**Task 1:** ~~Create a~~ **~~base~~** ~~class that will be used as the basis for the remainder of the lab.~~

1. ~~Create a new project. You can name this whatever you like.~~
2. ~~Design a class Game to abstractly model games. You may want to read the entire assignment before starting this task.~~
   1. ~~Include~~ **~~at least 2 attributes~~** ~~along with~~ **~~getters and setters~~** ~~for the attributes.~~
   2. ~~Create a default constructor and an overload constructor allowing you to set values for all attributes.~~
   3. ~~Define a~~ **~~virtual function called Play~~**~~. Define a~~ **~~non-virtual function called Winner~~** ~~that prints out the string~~ **~~“Not Yet”.~~**
   4. ~~Create the~~ ***~~implementation code for the above functions as required~~***~~. (for the milestone don’t write an implementation for Play yet, instead, make it pure virtual function “virtual void Play() = 0;”. After the milestone, write an implementation for Paly in game class. e.g. make it print out “what game?”)~~
3. Include in the submission ***how each member will be available*** in derived classes (i.e., not available, available if not overridden, etc.). Complete this before moving on to task 2.

**Task 2:** Create 2 classes that **inherit** from this class.

1. ~~Create a class for boardGame and video game that inherits from Game.~~
2. ~~Board game should have the following features:~~
   1. **~~Play~~** ~~should be defined as printing out~~ **~~“Roll the dice.”~~**
   2. **~~Winner~~** ~~should be defined as printing out~~ **~~“Dancing time.”~~**
   3. ~~Add~~ **~~1 additional attribute~~** ~~to the board game class. Include a~~ **~~getter and setter~~**~~.~~
3. ~~Video game should have the following features:~~
   1. **~~Play~~** ~~should be defined as printing out~~ **~~“Mash the buttons.”~~**
   2. **~~Winner~~** ~~should be defined as printing out~~ **~~“Winner music”~~**
   3. ~~Add~~ **~~1 additional attribute~~** ~~for the video game class (~~***~~different from the board game class~~***~~). Include a~~ **~~getter and setter~~**~~.~~
4. Include in the submission ***what version of the derived class members will be available*** in instances of the ***derived class*** and in instances of the ***derived class declared as the base*** class type. Complete this before moving on to task 3.

**~~Milestone:~~** ~~create a~~ **~~main~~** ~~function~~**~~,~~** ~~and inside it, try to~~ **~~create an object from game class~~**~~,~~ **~~take a screenshot of the error message~~**~~. instantiate two objects, one from each~~ **~~derived class~~**~~, and call~~ **~~Play~~** ~~and~~ **~~Winner~~** ~~functions for each object.~~

**Task 3:** Test the classes.

1. ~~Create a program that tests the classes.~~
   1. ~~Prompt the user for which class to create and values to set the attributes for that class. Include the option to create a board game or video game as a game. This may look like:~~

|  |
| --- |
| Press 1 for an instance of game.  Press 2 for an instance of board game.  Press 3 for an instance of video game.  Press 4 for an instance of a board game declared as a game  Press 5 for an instance of a video game declared as a game |

* 1. ~~Call the~~ **~~Play~~** ~~and~~ **~~Winner~~** ~~functions from the instance created in step a.~~
  2. ~~Create a~~ **~~Show~~** ~~function outside of classes that accepts a game as a parameter. This function should call the~~ **~~Play~~** ~~and~~ **~~Winner~~** ~~function of the input parameter.~~
  3. ~~Call the~~ **~~Show~~** ~~function from step c.~~
  4. ~~Ask the user if they wish to continue. If so, loop to step a.~~

1. ~~Use your test program to test all member functions and ensure the class is working correctly.~~
2. Include in the lab report a screenshot(s) of the output of a test. Include a discussion of how the actual results compared with the expected results from Task 2.

**Lab Submission:**

1. Write a lab report including the following information:
   1. ~~A description of the objectives/concepts explored in this assignment including why you think they are important to this course and a career in CS and/or Engineering.~~
   2. The sections from each task indicated to be included in the lab report.
2. ~~Include all source code from all tasks, input and output files (if any), and any special instructions to compile and run those programs.~~

**~~Lab Grading:~~**

1. ~~20% - Lab attendance~~
2. ~~15% - Task 1 has been correctly implemented and meets all requirements.~~
3. ~~20% - Task 2 has been correctly implemented and meets all requirements.~~
4. ~~25% - Task 3 has been correctly implemented and meets all requirements.~~
5. ~~20% - Lab report contains all required information and is well written.~~

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