**Instructions**

Based on the in-class Pong game we started to develop, you will complete the following:

* **Refer to Document Part3 and Part4 and complete the following**:
  + **Part 3 (2%):** Improve ball movement, make it start with a random direction and speed; Handle ball bouncing off top and bottom walls; Use simple AI to move AI paddle so it follows the ball
  + **Part 4 (1%):**Handle ball-paddle collision
  + When you complete these parts, the ai paddle must be controlled by the AI; you must be able to deflect the ball when the ball collides with either your paddle or AI paddle; the ball should bounce off the top and bottom walls; if the ball is missed either by you or AI, the ball must respawn back from the center.
  + Pay special attention to the variable names in the documents. Follow our naming convention set in the class even if the document follows a different convention (for example, the document has the ball's variable as 'Ball' whereas in our code, we named the ball as 'ball'. Likewise, we named aiPaddle and playerPaddle but the document names it differently. Also, the document mentions 'LoadGame()' function but that is referred in our code as 'InitGameWorld()' function. Likewise, they call it Draw)
* **Visual Improvements (0.5%) :** Add a center 'divider' rect and render it on screen to visually differentiate the player and AI's play area.
* **Improve AI Paddle (1.5%)**
  + Improve the AI paddle so the user has a chance to win yet keep it challenging
* **Scoring System (2%)**
  + Keep track of player score and AI score and display it on-screen. Initially, you can display it via an output to console window using cout. However, once I demonstrate in-class how to draw on screen using text, you will display using text.
* **Game-End Condition (1%)**
  + Determine how your game will end and handle the game-end condition.
  + You must explain in your submission about the your chosen game-end condition  (for example, first to score 5 points wins the game).
  + You should display a game over message (via console) and a summary of the scores (for example: Player wins the game. Final Score: 5-2).  Later, you will update this to display via text within the window.
* **System Architecture Design (1.5%)**
  + On paper, explain how you will improve the design of this game, using classes to improve the code structure. You do not have to actually implement it. We just want you to explain how you would do it. You can explain using UML diagrams. This can be submitted as a Word or PDF file.

**Submission (Following submission instructions:  0.5%)**

1. Game-end condition description (can add as comments in the submission)

2. Zip file of completed project

3. Screenshots of your game in action for the following test cases:

* Ball bounces off the top/bottom walls
* Ball bounces off the paddles
* Ball misses player paddle
* Ball misses AI paddle
* Scoring works
* Game-end condition works (i.e. the game over message with the final scores)

4. System Architecture Design doc (explaining how you will improve the system architecture by using classes)