Data Structures (2028C) -- Spring 2025 – Lab 2

Topics covered: Classes and multi-file programs

Lab due: ***Sunday, Feb 02 at 11:55PM for Monday Section***
Tuesday, Feb 04 at 11:55 PM for Wednesday Section

Objective:

The objective of this lab is to learn about classes and how to use their public variables and member functions.

Task 1: We are going to create a basketball game. Create a class for basketball player. It should include the following information:

- Name
- ShotsTaken
- ShotsMade
- PassesAttempted
- PassesMade

It should include the following class member functions:

- PassBall – this will return a bool value indicating if the pass was successful or resulted in a turnover. It will increment PassesAttempted and PassesMade accordingly. To determine the success of the pass, it will pull a random number between 1 and 100 and if the number is larger than the passing percentage (you need to calculate this) then the pass resulted in a turnover. Otherwise, it resulted in a successful pass.
- TakeShot – this will return the amount of points scored. The function will receive a value (1, 2, or 3) that the shot is worth as a parameter. It calculates if the shot scored it's points based on the shooting percentage (you need to calculate this) and the amount the shot is worth. It will pull a random number between 1 and 70 for a one point, 1 and 100 for a two point, and 1 and 125 for a three-point shot. If the random number is smaller than the shooting percentage, the player scored the points. This should also increment ShotsTaken and ShotsMade accordingly.

Create a main function. It should include an array of 5 players. It should initialize the players according to the attached list of possible players (you can hard code this in your program or allow your user to choose players).

The game will consist of each team getting 30 possessions. On the player's possession, they should be told which player has possession (display name and stats). The player with the ball is chosen randomly from the 5 players on the team. They may then choose to:

- Shoot – this will call that player's TakeShot function. The user will need to choose the value of the shot (1, 2, or 3). If the player misses, there is a 50% chance they retain possession (this additional possession doesn't count as one of the 30 in the game).
- Pass – this will call the player's PassBall function and if successful, the ball is transferred to the player indicated by the user.
- See player stats – This shows the stats for all the players on the team.
- See Score – this shows the current score and how many possessions remain in the game.

Upon losing possession, the opposing team will play. The opposing team will take a shot with 60% probability of making the shot and 50% probability of retaining possession if they miss. This should report each shot/rebound until they make a shot or fail to secure the rebound. Like with the user, rebounds here don't count as an additional possession.
You may choose to add additional functions to your class such as a function to print the player's stats.

Take screenshots of the output screen multiple times (doesn't need to be for the whole game but for considerable run of play for your team and the opposition's) during execution and include it in your lab report. Include in the submission your test files that include at a minimum a file shorter than, exactly, and longer than 10 lines.

Lab Submission:

1. Write a lab report including the following information:
    a. 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. Include screen shot(s) from Task 1.
    b. What public and private functions you created for your classes and in your main function and why.
2. Include all source code from all Tasks, input and output files, and any special instructions to compile and run those programs.
3. Package all files in a single zip folder and submit the file in canvas.

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.