Data Structures (2028C) -- Spring 2018 – Homework 2

Topics covered: Working with Classes

Homework due: ***Friday, Feb 16 at 1:20PM***

Objective:

The objective of this homework is to create classes and implement those classes in a simple game.

Scenario:

You are creating a new dice game. In this game, the house and the player each get a single die. The player will make a minimum bet then roll her die. The player can then look at her die and increase the wager by a maximum of double the original wager. The house will then roll its die. The player wins only if her die has a larger value than the house (house wins all ties). The game is over when the player chooses to stop or runs out of money. This game will run from a command line/console window. This needs to be written using C++.

Requirements:

1. Create a die class. This should have the ability to change the number of sides with a default of 6. It needs a roll method that will randomly assign a valid value.
2. Create a player class that includes an instance of a die class. The player should have a way to track their current amount of money.
3. Create a main method and any necessary support functions to enable a person to play the game until they lose or cash out. This should ask the player the number of sides on a die at the start of the game (minimum 6, maximum 20). This should include a function that accepts 2 die and a wager as parameters and returns the amount won (0 for a loss). This may require getters/setters in the die class to function correctly.
4. Bonus: Create a hard mode. This is an option the player selects when starting the game. The hard mode will increase the number of sides on the house's die every time the house loses and decreases the number of sides every time the house wins two turns in a row (minimum number of sides for house is the same number of sides as the player).

Submission:

Submit all source code files and any required data files in a zip file. Include a write up as a PDF including:

- The name of all group members (minimum 2 members, maximum 4 members).
- Instructions for compiling and running the program including any files or folders that must exist.
- What each group member contributed. If the contributions are not equitable, what portion of the grade each group member should receive.

Submission should be submitted via BlackBoard.

Grading:

1. 15% - Die class is declared and defined appropriately.
2. 15% - Player class is declared and defined appropriately.

3. 15% - Function in requirement 3 that accepts 2 die variables is declared and defined appropriately. Data in die that should remain private such as number of sides has appropriate getters/setters.
4. 45% - Program runs per scenario and requirements.
5. 10% - Bonus requirement works correctly.
6. 10% - Code is well formatted, well commented and follows a reasonable style.

If program fails to compile, the grade will be limited to a max grade of 50%. Note that with the bonus, the maximum grade possible is 110%.