DESKTOP APPLICATION DEVELOPMENT WITH JAVA – CEJV569

Assignment #1 **20% of your final grade**

Write a program that plays a dice game called C Raps. (Note: Any resemblance to the dice game craps is purely coincidental.) This game consists of 3 sub-games, any one of which you may choose to play.

Pass Line

In this game you roll the dice (called the come-out roll) and look at the total. If this total is equal to 7 or 11, then you win the amount of the bet. If the total is equal to 2, 3 or 12, then you lose the bet. Any other total of the dice becomes the "point", and you continue to roll the dice until you either (a) roll the point total again, in which case you win the amount of the bet, or (b) roll a total of 7, in which case you lose the amount of the bet. If the total of the dice is not 7 and not equal to the point, then you roll again, and continue to roll, until either condition (a) or (b) is met. Once the point has been determined, it does not change for subsequent rolls of the dice.

Field Bet

In this game you roll the dice just once. If this total is 3, 4, 9, 10 or 11, then you win the amount of the bet. If the total is 2, then you win double the bet. If this total is 12, then you win triple the bet. For any other total of the dice (5, 6, 7, or 8), you lose the amount of the bet.

Any 7

In this game you roll the dice just once. If this total is equal to 7, then you win quadruple the bet. For any other total of the dice, you lose the amount of the bet.

The user interface may be either console or GUI. To play the game you will need to set the initial value of the bank roll. A player can choose which game to play and the amount of a bet. To play the game means to roll the dice and decide if a winning or losing condition has occurred. Show the value of the two die and if the game is Pass Line then also show the point. After a game concludes you will show the current state of the bank roll. The game ends when either the player ends the program or their bank roll balance falls to zero.

Be sure to test the program thoroughly, including all possible error conditions that can occur, such as betting too much money or making a negative or non-integer bet. Display why a user wins or loses such as showing the results of winning on the come-out roll, losing on the come-out roll, winning by making the point, losing by failing to make the point, winning the field bet, losing the field bet, winning any 7, and losing any 7

Finally, remember that your solution must consist of classes that work together. Create a class for each game, a class that manages the money, a class for rolling the dice and a class for the presentation. Within the classes write methods that perform simple actions. Ideally each action should be contained in its own method.