DUE DATE:

Wednesday December 4, 2013, 11:59 pm

Project # 3

THE PROBLEM

You have just entered a casino and are looking for some entertainment. You notice a table where you can play a simplified version of the game called “craps”.

In this simplified version, a player rolls two dice. Each die has six faces. Each face contains 1,2,3,4,5 or 6 spots. After the dice have come to rest, the sum of the spots on the two top faces is calculated. If the sum is 7 or 11 on the first throw the player WINS. If the sum is 2, 3 or 12 on the first throw (called “craps”), the player LOSES (the “house” wins). If the sum is 4, 5, 6, 8, 9 or 10 on the first throw, that sum becomes the player’s “point”, To win, a player must continue rolling the dice until the player rolls the point value. However, the player loses if he or she rolls a 7 before rolling the point. If you win you WIN $1. If you lose you give away $1

Using classes from your previous projects you are to write a simulation of the game to determine whether you should risk your hard earned cash or not. You will run 10,000 simulated games to see what the odds are of winning at this game. If the odds are in the player’s favor, print out “Play this game” otherwise print out “Do not play this game”

Your program should print on the console the result of the first 5 simulated games and final statistics on the total number of games played, won and lost in the form of a histogram that displays the number of games won and lost. Finally you should have a final line stating the odds of winning and whether you should go ahead and play the game or not based on the results of your simulation.

Your program must ask the user of your program what seed they want to use and how many games they want to simulate.

You are given some latitude in how to display your results, but, essentially, your outputs should look something like this:

GAME 1 STARTED.

You rolled 3 and 4 for a total of 7. You WIN!

GAME 1 OVER

GAME 2 STARTED.

You rolled 1 and 1 for a total of 2. You LOSE!

GaME 2 OVER

GAME 3 STARTED.

You rolled 3 and 3 for a total of 6. The point is 6. Roll again

You rolled 3 and 5 for a total of 8. Roll again

You rolled 2 and 6 for a total of 8. Roll again

You rolled 1 and 6 for a total of 7. You LOSE!

GAME 3 OVER

GAME 4 STARTED

You rolled 3 and 3 for a total of 6. The point is 6. Roll again

You rolled 3 and 5 for a total of 8. Roll again

You rolled 2 and 6 for a total of 8. Roll again

You rolled 1 and 5 for a total of 6. You WIN!

GAME 4 OVER

GAME 5 STARTED

You rolled 3 and 4 for a total of 7. You WIN!

GAME 5 OVER

OVERALL STATISTICS:

Total games played 5000

Total games won: 2510

Total games lost: 2490

Display the histogram of Won/Lost games. To be consistent, a Loss is represented by the number 0 and a win is represented by the number 1.

Display the Conclusion:

Odds of winning: 50.2%

Conclusion: Play this game.

Guidelines:

You will need to make sure to use data hiding principles. Make sure

you use Public, Protected and Private access rights appropriately.

Not using data hiding principles will result in a 10% penalty

Make sure each class is declared and defined in separate header and

source files.

Not defining classes in separate files will result in a 50% penalty.

Make sure you have appropriate constructors and destructors.

Not having the appropriate constructors will result in a 10% penalty

Do not use a switch statement to update the counts in the histogram!

Using a switch statement in the updating method of your histogram

will result in a 30% penalty.

.

You will need to make sure you have selected the appropriate

constness for your methods. Not using the correct const attributes will

cost you 10% in penalty.

Your code MUST compile! If your code doesn’t compile you will

automatically receive a grade of 0.

Your executable MUST run. If not, you will automatically receive a

grade of 0.

If your code doesn’t display the histogram and counts appropriately

you will suffer a penalty of at least 50% depending on the cause of the

error.

You will submit your project through the digital drop box feature of

blackboard.

You will clean (Build -> Clean) the project (to minimize its size) , zip your project and submit the zipped file. Please name your zip file by lastname1\_lastname2\_2.zip for each group.

.

Each group only needs to submit one project file.

If you have any questions, please ask them!