

# COSC 201 – Assignment #3

## Fall 2011

**Objective:** Roll your own random number generator and implement two classic casino card games.

In class, you will learn about the standard random number generator used by most modern languages, the linear congruential generator. It's fast, simple and easy to implement. You'll not be using an LCG for this project.

Instead, you should choose one of the following random number generators to research and implement. Both of them have their advantages over the LCG and are used in a variety of applications. We will not be providing any details of these generators as it is up to your team to figure out how the generators work. The two random number generators that you can choose from are:

- 1.) Mersenne twister
- 2.) Acceptance-rejection method (be sure to choose an appropriate distribution!)

Then, using the random number generator that you have created, implement two casino card games:

- 1.) Blackjack (2 to 1 on Blackjack, insurance, dealer hits on soft 17, 5 deck shoe)
- 2.) Punto Banco Baccarat (5% commission, 8 to 1 on tie bets, 6 deck shoe)

You should provide all of the player options of these classic games and implement appropriate dealer actions/AI. For Baccarat it is important to list the last 50 results, in 5 rows of 10 results. As with the random number generators, we will not be discussing in class how these games are played, that is up to you and your team to research. I will answer any clarification questions on the games and the generators, including whether something should be included or not included as part of the games.

You should provide a way to track the monetary balance of the player and allow the player to switch games. The player should start out with \$1000 and balances should be saved to a file and used for the starting player balance in future executions of the program. You should also build in a reset option that resets the balance to \$1000.

**Expectations:** Your code will need to be neat, concise, well documented and above all, correct (see Testing). All classes should have headers and each method should have comments describing the method's function including pre and post conditions (see [www.cs.ucf.edu/~reinhard/classes/cop3503/notes01.pdf](http://www.cs.ucf.edu/~reinhard/classes/cop3503/notes01.pdf) for a good introduction to pre and post conditions). Any novel or possibly confusing code should be explained, as I do get confused and distracted easily.

Grading rubric will be given out at least a week ahead of the due date. You must work in teams of either 2 or 3 for this project. Once your team is set, have one person email me the composition of the team. You must do this by November 21<sup>st</sup> at 5pm or be randomly assigned to a team.

**Extra Credit:** Provide a GUI with card animations for both the player and the dealer. This is worth 15 points of extra credit if done to my satisfaction.

Implement a casino style craps game complete with animation and table. This is worth 25 points of extra credit if done to my satisfaction.

Yes, you can score over 100 points.

**DUE:** December 5, 11:59 pm Eastern via Digital Dropbox. Submit only one set of files per team.