CSC 330 – Artificial Intelligence
Programming Project #1 – The Game of NIM
Due Wednesday, February 7, by 5:00 PM

Justification:
This project is intended to be fairly simple, so that you can "ease" your way back into programming if you haven't done so in a while. Additionally, it will leave you with an implementation of the game of NIM, which you will modify and expand upon in future assignments.

Description:
This project should implement the game of NIM. For those who are unfamiliar with NIM, here are the rules for our version of the game:

- There are two players. Your program should have one human player (the user) and one computer player, and **the computer should always move first**.
- Between the two players is a pile of sticks (or other objects). Your program should start with 12 sticks, but the program should be flexible enough that the number of sticks could be easily changed.
- The players take turns (your program should allow the computer player to start all the time). On each turn, a player may take 1, 2, 3, or 4 sticks from the pile in the middle. Again, you should make your program flexible enough that you could change this rule to allow taking any number of sticks relatively easily.
- The winner of the game is the player that takes the last stick from the pile.

Your program does not need to have any graphics (although it can if you want to spend the extra time), but for each player's turn it should be obvious whose turn it is, how many sticks were taken, and how many sticks are left in the pile. When a player wins, there should be a clear message as to who won.

For this assignment, you may have the computer make any move you like on its turn. You can feel free to make the computer player as good as you like, though. You do need to make sure that the computer can only make legal moves; for example, if there are only 2 sticks left, it should not try to take 3. You should also make sure the human player makes only legal moves as well.

Programming Style:
We will be revisiting this game later in the course. It will make your lives a lot easier later on if you make it easy to change the procedure the computer uses to pick its move. I suggest making a function that computes the computer's move – that way if we change the method of game play, it will not require wholesale rewriting of the code. Also, you should write your code well (easy to read, well-commented, etc.) for the same reason.

Submission:
This program can be written in any programming language that you choose (if you choose a language other than C++, Java, or Python, please check with me first). Submit your source code file via the Moodle assignment page. Submit only the source file or files. You can also submit a file containing a link to a Replit project that you have shared with me. Let me know if you have any problems.

Specifications:
The project will be graded based on meeting the following specifications:

1. Your program represents a sincere effort to solve the problem.
2. Your program compiles and runs, assuming specification 1 is met.
3. Your program always has the computer make the first move.
4. Your program prevents the user from picking an inappropriate number of sticks.
5. Your program prevents the computer from picking an inappropriate number of sticks.
6. Your program makes it clear how many sticks are left after each move.
7. Your program successfully reports who the winner of the game is at the end.
8. The code is properly formatted/styled (indentations, comments, variable names, etc.)