The Game of NIM

Image taken from: https://wild.maths.org/play-win-nim

The game of NIM is a two-player game. In it, you start with a pile of stones. Players will then take turns taking 1,2, or 3 stones at a time. The goal is to be the last player to take all the stones.

In this project you will start with 16 stones. You will then create the game so that a player will play against a computer player.

You will need the following functions:
isValidMove(userMove: int) : bool
getUserMove(stones : int) : int
computerMove(stones : int) : int
checkWin(stones : int) : bool

The first thing you will need to do is create isValidMove(userMove: int) : bool. This function will have a Boolean return type and will take one parameter that is of type int. The purpose of this function is to test if the user's move is valid. In NIM, you can only take 1, 2, or 3 stones. However, you cannot take more stones than are in the pile. For example, if there were only 2 stones left, you cannot take 3. If the move is valid, you will return true else you will return false.

The next thing you will have to do is to create getUserMove(stones : int) : int. This function has a return type of int and takes one parameter that is also of type int. The purpose of this function is to get the user's move. You will first need to prompt the user to input how many stones they would like to take (1, 2, or 3). You will then need to call isValidMove() to see if what the user input is valid. If it is, you need to return the number of stones that the user took.

You will then need create computerMove(stones : int) : int. This function is where you will code the computer's moves. For the purposes of this project, you will need the computer to do the following given how many stones are left in the pile:

1) If the stone pile is 1 then the computer will take 1
2) If the stone pile is 2 then the computer will take 1
3) If the stone pile is 3 then the computer will take 2
4) If the stone pile is 4 then the computer will take 3
5) Otherwise, the computer will take a random number between 1 and 3

The last function you will create is checkWin(stones : int) : bool. This function has a return type of Boolean and only takes one parameter, the total number of stones. This function is very simple and can be coded in one line. It checks to see if there are any stones left. If there are none left, it returns true and the game ends, otherwise it returns false, and the game continues.

Once you have all the functions coded, you will need to code the main driver code. This is the code that organizes and calls your other functions. You will need to set the starting number of stones (16) and then make sure the game logic continues to execute until the checkWin() function returns true. This is also where you will need to kick off the user's turn and then the computers turn. Be sure to subtract the number of stones taken each turn from the total pile of stones.

Here is an example of how a game of NIM is played once your program is complete: