

# CS120 – Computer Science 1

## Lab #4

### Spring 2014

The purpose of this lab is to gain some experience using loop statements, namely `while`, `do-while` and `for`.

This week we expand the NIM program from week 1 with two additional features:

- Two-player games
- A visual representation of the game state

We will allow two players to compete against each other in a game of NIM, and we will show the user graphically how many tokens are left to be picked up by the player(s). Different from labs one through three, this lab is an excellent opportunity to practice pair programming. Therefore, we recommend that you work together in pairs for this lab, although you are also welcome to complete the lab on your own.

You are to write a program that presents both players with the choice to take between one and three tokens from the pile of remaining tokens, in alternating order. The code used in lab one readily accepts input from one player, and should be adapted to ask two users for input in an alternating order, where the second player replaces the computer player.

Additionally, at the end of each turn, the program should show how many tokens are left to be picked up in the game. The easiest way to do this is to print a character of your choice as many times as there are tokens left. For example, if a new NIM game starts with 24 remaining tokens and the first player picks up 2 tokens, the program should print  $24 - 2 = 22$  characters to let the players know there are 22 remaining tokens. This step takes place at the end of each turn.

Below is an example execution of the new version of NIM with two players and the visual output as described above:

```
-bash-4.1$ ./a.out
Player 1 enter your move (1-3): 3
20 objects remaining.
#####
Player 2 enter your move (1-3): 3
17 objects remaining.
#####
Player 1 enter your move (1-3): 2
15 objects remaining.
#####
Player 2 enter your move (1-3): 3
```

```
12 objects remaining.
#####
Player 1 enter your move (1-3): 3
9 objects remaining.
#####
Player 2 enter your move (1-3): 3
6 objects remaining.
#####
Player 1 enter your move (1-3): 2
4 objects remaining.
####
Player 2 enter your move (1-3): 3
1 objects remaining.
#
Player 1 enter your move (1-3): 1
0 objects remaining.
```

Player 0 wins!!!

Press enter to quit.

When you have written a working program, hand in a hardcopy of both the source code and a typescript that shows a complete playthrough of a two-player game. Also submit your source code via `cscheckin`. If you did the program in pairs, you can turn in one copy for both of you – just be sure that both names are included in the files.