

## Programming Project 02

Edit: (on 9/6) numbering of piles made consistent as “1” and “2” (instead of 0 and 1).

This assignment is worth 20 points (2% of the course grade) and must be **completed and turned in before 11:59 PM on Monday, September 17, 2018.**

### Assignment Overview

(learning objectives)

This assignment will give you more experience on the use of:

1. integers (int)
2. conditionals
3. iteration
4. input/output

The goal of this project is to play a simplified game of Nim.

### Assignment Background

Nim is a mathematical game of strategy in which two players take turns removing objects from distinct piles. The game will be played between the human player and the computer player. In this project, you will handle the computer player while a user plays the game. Choosing the optimal game strategy is complicated and will likely require you to do things that we have not covered yet. We will use a simplified version here.

The rules: “ Given two piles, numbered 1 and 2, in which each pile contains some numbers of stones. In each turn, a player can choose only one pile and remove one, two, or three stones from that pile. The player who cannot move is considered to lose the game (i.e., the one who takes the last stone is the winner). ”

### Project Description

Your program must meet the following specifications. Note that control gets complex to play this game: (one instructor used 4 `while` loops and 7 if statements).

The program begins by offering a welcome message and display the rules of the game. Then, asking the player if he wants to play the game. That control is provided in the starting code provided.

1. Initialize both piles to have 5 stones each.
2. You will need a `while` loop to play the game (this will be in addition to the while loop in the provided code that is used to ask whether the human player wants to play another game.) What Boolean expression should control this loop? Hint: think about the size of piles.
3. Play alternates between the human and computer. Use a Boolean variable to keep track of who is playing. The human player starts.
4. Within this `while` loop:
  - a. When it is the human player's turn:
    - i. First prompt for the pile (1 or 2). Continue prompting until valid pile number is entered. A valid pile is non-empty and numbered 1 or 2.
    - ii. Next prompt for a number of stones to be removed from that pile. Continue prompting until a valid number of stones is entered.
  - b. When it is the computer's turn:

- i. Select the pile that the human player did NOT use on their turn.  
If the selected pile is empty, select the other pile, i.e. the one the human player used on their turn (note that if both piles are empty the game should have already ended).
  - ii. The number of stones to be removed by the computer is always 1 stone.
- c. Then, remove the stones from the pile. Display the number of stones in each pile after removal. Remember to switch players.
- 5. The player who takes the last stone is the winner.
- 6. When the game has come to its conclusion, the program should output whether the human or the computer won. It then should display the cumulative record and ask the user whether to play again. (And if the user selects to play again, your program should.)
- 7. If the user choose not to play the game again, the program should display a goodbye message.
- 8. Your program should verify that each of the user's moves is valid. (That is, the pile must be legal, the user must remove between 1 and 3 stones, and the user must not remove more stones than the pile holds.) If the user's move is illegal, your program should repeatedly print a sensible error message and get a new input, until the user selects a legal move.

## Assignment Deliverable

The deliverable for this assignment is the following file:

`proj02.py` – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the **Mimir** before the project deadline.

## Assignment Notes

1. To clarify the project specifications, sample output is appended to the end of this document.
2. Items 1-6 of the Coding Standard will be enforced for this project.
3. We provide a `proj02.py` program for you to start with. It has a simple while loop (notice how input is prompted before the loop and at the bottom of the loop).
4. You are not allowed to use advanced data structures such as list, dictionaries, classes....
5. You do not need to check for any input errors other than those mentioned in this description.

**If you wanted to create a more fun game (note that such a game wouldn't pass Mimir tests!),** you could `import random`, assign stones to piles randomly using something such as `random.randint(3, 5)`, have the computer choose a pile randomly and choose stones randomly using `random.randint`.

## Test Cases

### Test 1

Welcome to the game of Nim! I'm probably going to win...

Nim is a simple two-player game based on removing stones.

The game begins with two piles of stones, numbered 1 and 2.

Players alternate turns. Each turn, a player chooses to remove one, two, or three stones from some single pile. The player who removes the last stone wins the game.

Would you like to play? (0=no, 1=yes) 0

Thanks for playing! See you again soon!

## Test 2

Welcome to the game of Nim! I'm probably going to win...

Nim is a simple two-player game based on removing stones.

The game begins with two piles of stones, numbered 1 and 2.

Players alternate turns. Each turn, a player chooses to remove one, two, or three stones from some single pile. The player who removes the last stone wins the game.

Would you like to play? (0=no, 1=yes) 1

Start --> Pile 1: 5      Pile 2: 5

Choose a pile (1 or 2): 1

Choose stones to remove from pile: 3

Player -> Remove 3 stones from pile 1

Pile 1: 2      Pile 2: 5

Computer -> Remove 1 stones from pile 2

Pile 1: 2      Pile 2: 4

Choose a pile (1 or 2): 2

Choose stones to remove from pile: 3

Player -> Remove 3 stones from pile 2

Pile 1: 2      Pile 2: 1

Computer -> Remove 1 stones from pile 1

Pile 1: 1      Pile 2: 1

Choose a pile (1 or 2): 1

Choose stones to remove from pile: 1

Player -> Remove 1 stones from pile 1

Pile 1: 0      Pile 2: 1

Computer -> Remove 1 stones from pile 2

Pile 1: 0      Pile 2: 0

Computer wins!

Score -> human: 0 ; computer: 1

Would you like to play again? (0=no, 1=yes) 0

Thanks for playing! See you again soon!

## Test 3

Welcome to the game of Nim! I'm probably going to win...

Nim is a simple two-player game based on removing stones.

The game begins with two piles of stones, numbered 1 and 2.

Players alternate turns. Each turn, a player chooses to remove one, two, or three stones from some single pile. The player who removes the last stone wins the game.

Would you like to play? (0=no, 1=yes) 1

Start --> Pile 1: 5      Pile 2: 5

Choose a pile (1 or 2): 3

Pile must be 1 or 2 and non-empty. Please try again.

Choose a pile (1 or 2): 1

Choose stones to remove from pile: 3

Player -> Remove 3 stones from pile 1

Pile 1: 2      Pile 2: 5

Computer -> Remove 1 stones from pile 2

Pile 1: 2      Pile 2: 4

Choose a pile (1 or 2): 1

```

Choose stones to remove from pile: 2
Player -> Remove 2 stones from pile 1
Pile 1: 0    Pile 2: 4
Computer -> Remove 1 stones from pile 2
Pile 1: 0    Pile 2: 3
Choose a pile (1 or 2): 2
Choose stones to remove from pile: 2
Player -> Remove 2 stones from pile 2
Pile 1: 0    Pile 2: 1
Computer -> Remove 1 stones from pile 2
Pile 1: 0    Pile 2: 0

Computer wins!
Score -> human: 0 ; computer: 1

Would you like to play again? (0=no, 1=yes) 0

Thanks for playing! See you again soon!

```

## Test 4

```

Welcome to the game of Nim! I'm probably going to win...
Nim is a simple two-player game based on removing stones.
    The game begins with two piles of stones, numbered 1 and 2.
    Players alternate turns. Each turn, a player chooses to remove one,
    two, or three stones from some single pile. The player who removes the
    last stone wins the game.
Would you like to play? (0=no, 1=yes) 1
Start --> Pile 1: 5    Pile 2: 5
Choose a pile (1 or 2): 1
Choose stones to remove from pile: 1
Player -> Remove 1 stones from pile 1
Pile 1: 4    Pile 2: 5
Computer -> Remove 1 stones from pile 2
Pile 1: 4    Pile 2: 4
Choose a pile (1 or 2): 1
Choose stones to remove from pile: 1
Player -> Remove 1 stones from pile 1
Pile 1: 3    Pile 2: 4
Computer -> Remove 1 stones from pile 2
Pile 1: 3    Pile 2: 3
Choose a pile (1 or 2): 1
Choose stones to remove from pile: 1
Player -> Remove 1 stones from pile 1
Pile 1: 2    Pile 2: 3
Computer -> Remove 1 stones from pile 2
Pile 1: 2    Pile 2: 2
Choose a pile (1 or 2): 1
Choose stones to remove from pile: 1
Player -> Remove 1 stones from pile 1
Pile 1: 1    Pile 2: 2
Computer -> Remove 1 stones from pile 2
Pile 1: 1    Pile 2: 1
Choose a pile (1 or 2): 1
Choose stones to remove from pile: 1
Player -> Remove 1 stones from pile 1
Pile 1: 0    Pile 2: 1
Computer -> Remove 1 stones from pile 2
Pile 1: 0    Pile 2: 0

Computer wins!

```

```
Score -> human: 0 ; computer: 1

Would you like to play again? (0=no, 1=yes) 1
Start --> Pile 1: 5      Pile 2: 5
Choose a pile (1 or 2): 1
Choose stones to remove from pile: 3
Player -> Remove 3 stones from pile 1
Pile 1: 2      Pile 2: 5
Computer -> Remove 1 stones from pile 2
Pile 1: 2      Pile 2: 4
Choose a pile (1 or 2): 1
Choose stones to remove from pile: 2
Player -> Remove 2 stones from pile 1
Pile 1: 0      Pile 2: 4
Computer -> Remove 1 stones from pile 2
Pile 1: 0      Pile 2: 3
Choose a pile (1 or 2): 1
Pile must be 1 or 2 and non-empty. Please try again.
Choose a pile (1 or 2): 2
Choose stones to remove from pile: 3
Player -> Remove 3 stones from pile 2
Pile 1: 0      Pile 2: 0

Player wins!
Score -> human: 1 ; computer: 1

Would you like to play again? (0=no, 1=yes) 0

Thanks for playing! See you again soon!
```

## Grading Rubric

Computer Project #02

Scoring Summary

General Requirements:

( 5 pts) Coding Standard 1-6

(descriptive comments, mnemonic identifiers, format, etc...)

Implementation:

(2 pts) Test Case 1:

(4 pts) Test Case 2:

(4 pts) Test Case 3:

(5 pts) Test Case 4: