CS136L: Computer Science II Lab – Fall 2018

Assignment	Points	Announced	Due
#3	40	Sept-17	Sept-28

The Game of Nim

1 Overview

In this lab, you will work on implementing a flavor of the famous mathematical strategy game: Nim.

2 Learning Outcomes

By the end of this project students should be able to:

- write, save, and evaluate simple programs;
- read and write programs with numerical literals;
- read and write programs with simple function calls (e.g., input, print);
- read and write programs with conditionals;
- read and write programs with loops;
- break up simple problems into multiple steps;
- work effectively with a partner using pair-programming;
- write an effective report that describes the students' problem-solving process.

3 Pre-Lab Instructions

Do this part before you come to lab:

- Study chapter 6 from the textbook. Pay special attention to section 6.9.1 about random numbers generation.
- Watch this video about the game Nim: https://www.youtube.com/watch?v=Hof7P__P68I
- Read this Wikipedia article about the game Nim: https://en.wikipedia.org/wiki/Nim

4 Submission Instructions

1. Use a basic code editor (e.g. Notepad, Notepad++, Atom…) and not an IDE for this lab assignment.
2. In addition to the lab report, submit a Java file named `Nim.java`.
3. Don't submit .class files nor .jar files nor archives (e.g. .zip, .rar …)

5 Lab Instructions

Do this part in lab:

The game of Nim

This is a well-known game with a number of variants. The following variant has an interesting winning strategy. Two players alternately take marbles from a pile. In each move, a player chooses how many marbles to take. The player must take at least one but at most half of the marbles. Then the other player takes a turn. The player who takes the last marble loses.

Write a program in which the computer plays against a human opponent. Generate a random integer between 10 and 100 to denote the initial size of the pile. Generate a random integer

between 0 and 1 to decide whether the computer or the human takes the first turn. Generate a random integer between 0 and 1 to decide whether the computer plays smart or stupid. In stupid mode the computer simply takes a random legal value (between 1 and n / 2) from the pile whenever it has a turn. In smart mode the computer takes off enough marbles to make the size of the pile a power of two minus 1—that is, 3, 7, 15, 31, or 63. That is always a legal move, except when the size of the pile is currently one less than a power of two. In that case, the computer makes a random legal move.

You will note that the computer cannot be beaten in smart mode when it has the first move, unless the pile size happens to be 15, 31, or 63. Of course, a human player who has the first turn and knows the winning strategy can win against the computer. Below is an example of a game run:

```
Current number of marbles in pile: 95
Computer removes 12 marble(s).
Current number of marbles in pile: 83
How many marbles do you want to remove:
12
Current number of marbles in pile: 71
Computer removes 17 marble(s).
Current number of marbles in pile: 54
How many marbles do you want to remove:
20
Current number of marbles in pile: 34
Computer removes 6 marble(s).
Current number of marbles in pile: 28
How many marbles do you want to remove:
28
How many marbles do you want to remove:
25
How many marbles do you want to remove:
20
How many marbles do you want to remove:
12
Current number of marbles in pile: 16
Computer removes 4 marble(s).
Current number of marbles in pile: 12
How many marbles do you want to remove:
12
How many marbles do you want to remove:
4
Current number of marbles in pile: 8
Computer removes 4 marble(s).
Current number of marbles in pile: 4
How many marbles do you want to remove:
6
How many marbles do you want to remove:
3
How many marbles do you want to remove:
```

```
2
Current number of marbles in pile: 2
Computer removes 1 marble(s).
Current number of marbles in pile: 1
How many marbles do you want to remove:
1
User took last marble. Computer wins!
```

6 Lab Report

Each pair of students will write a single lab report together and each student will turn in that same lab report on BBLearn. Submissions from each student on a pair should be identical.

Your lab report should begin with a preamble that contains:

- The lab assignment number and name
- Your name(s)
- The date
- The lab section number

It should then be followed by four numbered sections:

1. Problem Statement

In this section you should describe the problem in *your* own words. The problem statement should answer questions like:

- What are the important features of the problem?
- What are the problem requirements?

This section should also include a reasonably complete list of requirements in the assignment. Following your description of the problem, include a bulleted list of specific features to implement. If there are any specific functions, classes or numeric requirements given to you, they should be represented in this bulleted list.

2. Planning

In the second section you should describe what planning you did in order to solve the problem. You should include planning artifacts like sketches, diagrams, or pseudocode you may have used. You should also describe your planning process. List the specific data structures or techniques you plan on using, and why.

3. Implementation and Testing

In the third section you should describe how you implemented your plan. As directed by the lab instructor you should (as appropriate) include:

- a copy of your source code (Submitted in BBLearn as .java files)
- a screen shot of your running application / solution
- results from testing

4. Reflection

In the last section you should reflect on the project. Consider different things you could have done to make your solution better. This might include code organization improvements, design improvements, etc.

You should also ask yourself what were the key insights or features of your solution? Were there alternative approaches or techniques you could have employed? How would these alternatives have impacted a different solution?

5. Partner Rating

Every assignment you are required to rate your partner with a score -1, 0 or +1. This should be submitted in the comment section of the BBLearn submission, and not in the report document. If you don't want to give your partner a negative rating making sure not to use a dash before listing the number! You do not have to tell your partner the rating you assign them. A rating of 1 indicates that your partner was particularly helpful or contributed exceptional effort. A rating of 0 indicates that your partner met the class expectations of them. Rating your partner at -1 means that they refused contribute to the project, failed to put in a reasonable effort or actively blocked you from participating. If a student receives three ratings of -1 they must attend a mandatory meeting with the instructor to discuss the situation, and receiving additional -1 ratings beyond that, the student risks losing a letter grade, or even failing the course.

7 Grading Rubric

This lab assignment will be graded according to this rubric:

Criteria / Component	Points
Report	10 pts
Implementing stupid mode	10 pts
Implementing smart mode	20 pts

Lab assignment penalties:

Item	Points
Missing names	-4
Missing partner rating	-2
Missing screenshots	-2
Insufficient / No commenting	-2 / -4
Too late to pair (if attended)	-4
Absent	-12
Non-compiling program	-40

Note:

If your partner is not responding to your emails and/or not collaborating, don't hesitate to reach out to the lab TA aide and/or the primary instructor.