# Homework 5: Towers of Hanoi **cs244**

## Due: October 14, 8:00 AM

*Allowances, if any, for late work and related penalties will be as described in the syllabus.*
*As always, check with the instructor for any amendments or changes.*

**Overview**

Implement a program to recursively solve the Hanoi problem which also displays the total number of recursive calls made.

Topics: Recursion, C++ classes, conditional cases, code testing, peer reviews

**Preparation**

**Step 1: Create a Folder**

In Linux in your …/Documents/Programs folder
Create a folder named ***A05***
This is necessary as you will compress the A05 folder
and its contents for submission.

**Step 2: Check D2L**

There may be starter code for this assignment available on D2L.

**Files to Create**

The files for this assignment MUST be named:

**hanoi.h hanoi.cpp hanoiMain.cpp README.txt**

Your *main* function is to be located in hanoiMain.cpp.

All code files should have comments with the file name, your name, the last modified date, and a description of the file contents. Each function should be appropriately commented in the header and implementation files – most should include pre and post conditions. The body of each function may need comments for the less than obvious parts of the code (if any).

**The final submission of files should include a file named: README.txt**

It provides instructions on how to compile/link your program (EX: `g++ file1.cpp file2.cpp`)
It also provides instructions on how to run your program (EX: after compile type: `./a.out`)

**makefile**

This assignment does not *require* a makefile but you can provide one
or *for this assignment* you may apply the following

To Compile at the Command Line Type:

```
g++ hanoi.cpp hanoiMain.cpp
```

To Run the executable created at the Command Line Type: ./a.out

# Homework 5: Towers of Hanoi cs244

## Assignment Description

This assignment is derived from content found in Data Structures Using C++ 2$^{nd}$ Edition by D.S. Malik. Understanding the content of chapter 6 of that book as well as an examination of the code contained therein will be helpful in completing this assignment.

Create a C++ class named Hanoi with a public function name ***SolveHanoi*** to recursively solve the Hanoi Problem. The function should display all the moves in the form:

Move disk [disk id] from [needle id] to [needle id]

Where

[disk id] is 1 for the smallest and N for the largest (assuming N disks)

[needle id] is one of three strings: “left” “middle” “right”

After the last move is made, the public function, ***SolveHanoi*** should display the total number of recursive calls that were made in the format:

cout << "Total recursive calls = " << *m_recurseCount* << endl;

The solution must be designed with 2 base cases: One for a tower of 1 disk and one for a tower of 2 disks. Thus for these two cases the number of recursive calls should be 1 (the function is called 1 time). For cases of 0 or negative number of disks the recursive call count should be 0. For testing 3 disks should need 3 total recursive calls and 4 disks should require 7 calls.

Write a main function in the file hanoiMain.cpp to test your code.

Might want to test what happens if the “left” needle is not the initial start needle

***SolveHanoi*** should take 4 parameters in the order given below:

int count, string needle1, string needle3, string needle2

count = number of disks to move<br>
needle1 = the “from” needle<br>
needle3 = the “to” needle<br>
needle2 = the “intermediate use” needle

## Hints

Code for one of your class functions could/should be based on the following:

- Base Case: first needle contains only one disk
  - Move disk directly from needle 1 to needle 3
- Base Case: first needle contains only two disks
  - Move first disk from needle 1 to needle 2
  - Move second disk from needle 1 to needle 3
  - Move first disk from needle 2 to needle 3
- General Case: first needle contains three or more disks
  - Make recursive calls as needed

# Homework 5: Towers of Hanoi

cs244

## Tester File

Design your main function, in the hanoiMain.cpp file, to demonstrate and test the capabilities of your Hanoi class.

## Grading

100 points possible

Assuming it is turned in on time:
Your program will be peer reviewed and evaluated by members of your class.
Each will assign a grade to it from 0 to 100.
The instructor will also evaluate your work, scoring it from 0 to 100.
The final grade will then be assigned by the instructor based on the three scores.

Grading will be based on compilability, runability, correctness, comments, programming style, and similar attributes. Further details may be given in class.

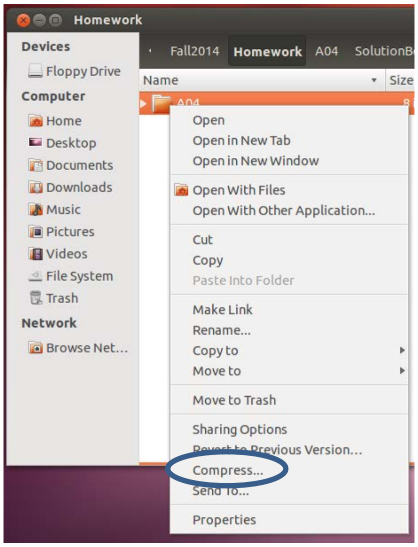
## Turn-In Directions

Correctly submitting your work is worth 0 points, BUT
if not done correctly will likely result in nothing to grade.

### Preparation

In Ubuntu Linux browse to your A05 folder
Make sure your source code files are in the folder

**Right click on the A05 folder,**
**Select Compress…**



**Set the file name to be A05_*yourlastname***.tar.gz
where *yourlastname* is your last name

Example: if your last name is Gollygee
then the filename would be
A05_Gollygee.tar.gz

**Click on the Create Button**
This should create the file named A05_ *yourlastname*.tar.gz

### Submit

Submit the A05_ *yourlastname*.tar.gz file
to the correct course drop box in D2L

**Peer Evaluations**

For peer review evaluations you should follow the instructions given in class.
In most cases this will involve giving the above compressed files to
the peers assigned to evaluate your program.
Again in most cases, they will then review and evaluate your program.
This often means your code will need to function with a test program they created.
This emphasizes the importance of coding to specification and doing so in a readable and maintainable manner.