

# LAB 1 (JAVA REFRESHER)

---

CSC 172 (Data Structures and Algorithms)

Spring 2018

University of Rochester

**Due Date: In-Lab Demo**

## Introduction

The labs in CSC172 will follow a pair programming paradigm. Every student is encouraged (but not strictly required) to have a lab partner. Labs will typically have an even number of components. The two partners in a pair programming environment take turns at the keyboard. This paradigm facilitates code improvement through collaborative efforts and exercises the programmer's cognitive ability to understand and discuss concepts fundamental to computer programming. The use of pair programming is optional in CSC172. It is not a requirement. You can learn more about the pair programming paradigm, its history, methods, practical benefits, philosophical underpinnings, and scientific validation at [http://en.wikipedia.org/wiki/Pair\\_Programming](http://en.wikipedia.org/wiki/Pair_Programming).

This lab has only two problems. You and your partner(s) should switch off typing each problem as explained by your lab TA. As one person types the lab, the other should be watching over the code and offering suggestions. You need to create two different projects for the task. Do not erase any previous work when you switch.

## Tasks

1. Implement a method `isAnagram` which takes two strings as input and decides if those strings are anagrams or not. An anagram is a word, phrase, or name formed by rearranging the letters of another, such as cinema, formed from iceman.

Implement the `main` method which takes two string as input from the user and prints whether the strings are anagrams.

2. Implement a method `isRotation` which takes two strings as input and decides if one string is a rotation of the other. For example, 'cdeab' is a rotation of 'abcde'.

Implement the `main` method which takes two string as input from the user and prints whether one string is the rotation of the other.

## Demo

After each task, show your output to the Lab TAs. They will test your code and grade you for correctness. Before you leave the lab, make sure your work is graded. You can demo your output on either session (Monday/Wednesday) or (Tuesday/Thursday).

## Submission and Demo

Hand in the source code from this lab at the appropriate location on the Blackboard system at `learn.rochester.edu`. You should hand in a single zip (compressed archive) `Lab1.zip` containing your source code and a README file, as described below.

- A plain text file named README that includes your contact information, your partner's name, a brief explanation of the lab (a one paragraph synopsis. Include information identifying what class and lab number your files represent.), and one sentence explaining the contents of any other files you hand in. Also, include the compile and run steps.

## Grading:

Total Points: 100  
50 points for each problem.

## Notes:

All labs are open book. You can get code snippets from the internet if you need to (make sure you cite those properly). But that is not the purpose. We want you to sit together, think about an algorithm, and then implement it together with your partner (that sounds fun! isn't it?)

## Special Notes:

Please read carefully:

Lab 1 is only lab where demonstrating how your code works is mandatory. Also, this is the only lab where TAs will NOT grade your submitted code, rather the grading will be done solely based on your demo.

## Frequently asked questions (FAQ)

- **Q: Why do we need to submit our code if it's is never graded?**
- A: All future labs and projects will be graded based on your submission on Blackboard. We are mandating the submission just to make sure that you know how to submit.
- **Q. My Lab partner has submitted the code. Do I need to submit again?**
- A. Yes. Even though you are working as a team, each team member must submit his/her code separately for any credit. This is true for all future labs and projects too.