Developer Interview Project

College Course Prerequisites

**Description**

A local college has asked you to develop a program to determine which order new students should take their courses in. Some courses have prerequisites which need to be taken first. The college will provide you with a list of their courses and prerequisites, and would like you to output a class schedule of which order the courses should be taken in.

**Requirements**

1. The program should accept an array of strings defining courses. Each string contains the name of a course followed by a colon and space, then any prerequisites required by that package. For simplicity we’ll assume a course can have at most one prerequisite.
2. Ensure the project will compile at your final check in.
3. Ensure the project have a way to enter data either from a console or UI, to validate the class schedule.
4. The program should output a comma separated list of course names in the order they should be taken, such that a course’s prerequisite will always precede it.
5. The program should reject as invalid any circular dependencies, such as when course A requires course B, which requires course A.
6. You can complete the exercise in C#.
7. Use Test Driven Development (TDD) and include your unit tests in your submission.
8. Submit your code in a git repository (emailed in a zip file). Commit throughout the process; we want to see multiple commits so we can see your progress as you code the solution.

For example, the input

Advanced Pyrotechnics: Introduction to Fire

Introduction to Fire:

represents two packages, “Advanced Pyrotechnics” and “Introduction to Fire”, where “Advanced Pyrotechnics” depends on “Introduction to Fire”. In this case the output should be Introduction to Fire, Advanced Pyrotechnics indicating that Introduction to Fire, needs to be taken before Advanced Pyrotechnics.

**Example of valid input**

Introduction to Paper Airplanes:

Advanced Throwing Techniques: Introduction to Paper Airplanes

History of Cubicle Siege Engines: Rubber Band Catapults 101

Advanced Office Warfare: History of Cubicle Siege Engines

Rubber Band Catapults 101:

Paper Jet Engines: Introduction to Paper Airplanes

A valid output for the above would be:

Introduction to Paper Airplanes, Rubber Band Catapults 101, Paper Jet Engines, Advanced Throwing Techniques, History of Cubicle Siege Engines, Advanced Office Warfare

**Example** **of input that should be rejected (contains cycles)**

Intro to Arguing on the Internet: Godwin’s Law

Understanding Circular Logic: Intro to Arguing on the Internet

Godwin’s Law: Understanding Circular Logic

**Review**

Please submit your code in a git repo (zipped and emailed, not on github) where you have committed throughout the process so that we can see your progress as you code the solution. We will be running your code against our internal set of tests. We’ll get back to you with next steps after we review your submission.