The solution is implemented with Java8 utilizing the stream API, Static method referencing and making the class operate single responsible manner to reduce the coupling with additional logics. This helps to write test cases with Junit4 against the classes.

The starting point of the class is in *CourseOrderOrganizer* class there the files are looked up and passed to a *CourseDependencyHolderFacade* to build up course dependency map.

The Façade uses *FileProcessingService* as a service to open the files (course.csv and prerequisites.csv) and build the dependency map.

Once the structure is build, the map is passed to *CourseDependencyOrganizingService* to groom the map to find the course sequence.

I have noted the course stuck in the infinite loop where the course 4 in unable to complete as it caused deadlock:

4 -> 9 -> 5 -> 4

To make the process simple I have removed the course 5 from depending on 4.

**Test Cases:**

I have implemented Junit test on *CourseDependencyHolderFacadeTest* and *CourseDependencyOrganizingServiceTest*

**Improvements:**

1. Can use logging instead of System.out
2. Use Spring to auto wire rather than constructor creation.