Course Scheduling

Problem statement | Input / Output | Badges and score

Build instructions | Help Center

Next step:

You have attempted this problem in c# and scored 85. Read our help docs and resubmit code to achieve a higher score.

Starter Kit

View Portfolio

Context

The head of the Learning management system (LMS) at Intuit has hired you as a consultant. The LMS team has the goal of upskilling the employees with the latest topics via courses. You need to help build a system to schedule and manage the courses.

Goal

Your job is to build a simple command line application, which does the following:

- Add course offering
  - A course offering has course title, instructor and date.
  - It should also contain a minimum & maximum number of employees for the course offering.
- Register for the course
  - Employees can register for the courses.

- If no. of employees registered for the course has reached the maximum, the result will be COURSE_FULL_ERROR.
- Otherwise, result of registration will be ACCEPTED.

- Cancel registration
  - Employees can cancel their registration until the course allotment is completed.
- Course allotment
  - This feature allots employees to course offering, before the course offering date.
  - It should print a list of all the employees with their details along with their final course allotment status (Registration Number, Employee Name, Email, Course Offering ID, Course Name, Instructor, Date, Final Status). The list should be sorted based on the Registration number.
  - If sufficient registrations are not received then the course offering itself gets cancelled.
  - The employees who have registered will get confirmed unless the minimum number of registrations is not received.
  - Even if the course offering gets canceled due to the minimum number of employees not registered, the list should be printed.

Commands

Every input command has an output. The format is as given
‹COMMAND› ‹parameter-1›...‹parameter-n› : ‹OUTPUT›

Add course offering

COMMAND	PARAMETERS	OUTPUT
ADD-COURSE-OFFERING	‹course-name› ‹instructor› ‹date-in-ddmmyyyy› ‹minEmployees› ‹maxEmployees›	‹course-offering-id›

The format of course-offering-id is OFFERING-‹COURSE-NAME›-‹INSTRUCTOR›

Register for the course

COMMAND	PARAMETERS	OUTPUT
REGISTER	<email-id> <course-offering-id>	<course-registration-id> <status>

- The combination of email-id and course-offering-id in the input should be unique
- The format of course-registration-id is REG-COURSE-<EMPLOYEE-NAME>-<COURSE-NAME>
- If number of employees has not exceeded the maximum number of employees allowed for the course offering, status will be ACCEPTED
- If number of employees has exceeded the maximum number of employees allowed for the course offering, status will be COURSE_FULL
- If the minimum number of employees for the course offering is not reached before the course date, the status of the course offering would be COURSE_CANCELED
- Course-registration-id will only be returned if the status is ACCEPTED

Course allotment

COMMAND	PARAMETERS	OUTPUT
ALLOT-COURSE	<course-offering-id>	<course-registration-id> <email-id> <course-offering-id> <course-name> <instructor> <date-in-ddmmyyyy> <status>

The output should be sorted by course-registration-id in ascending order

Cancel registration

The employee can cancel the course registration given a course registration id until course allotment

COMMAND	PARAMETERS	OUTPUT
CANCEL	<course-registration-id>	<course-registration-id> <status>

- There are 2 statuses : CANCEL_ACCEPTED, CANCEL_REJECTED
- CANCEL_ACCEPTED when the cancellation is successful.
- CANCEL_REJECTED when the course is already allotted.

Assumptions

- If there is validation error in the input (data validation or mandatory fields missing) then it should print an error message INPUT_DATA_ERROR.
- Employees can only cancel their registration until the course is allotted.
- Instructor names are unique
- Course names are unique
- None of the input fields accept whitespace (whitespace acts as a delimiter between fields)
- Course offering ID generated is a combination of OFFERING-<COURSENAME>-<INSTRUCTORNAME>
- Registration ID generated is a combination of REG-COURSE-<EMPLOYEENAME>-<COURSENAME>
- <EMPLOYEENAME> is extracted from email ID: everything before the @ sign in the email