

Assignment - Team and Leaves Management

[Weight: 12 marks out of the final mark of this course]

Deadline: April 25, 2016 (Monday of revision week)

For late submissions, 2% of your original marks will be deducted if you hand in 1-day late (i.e. on Apr 26), 4% for 2-days (i.e. on Apr 27), 8% for 3-days (i.e. on Apr 28), 16% for 4-days (i.e. on Apr 29), 32% for 5-days (i.e. on Apr 30), 64% for 6-days (i.e. on May 1). Assignments handed on or after May 1 will get zero mark.

Academic dishonesty is strictly prohibited. The principle concerns whether students get their deserved marks and do not intend to cause unfairness. Dishonesty also involves when one let others have a chance to copy his/her code,

Students must obtain the following results in sequence:

Phase 1 (100% correct in PASS) ==> Phase 2 (100% correct in PASS) ==> Phase 3(100% correct in PASS)

- If you can finish Phase 1 with good programming styles + OO programming skills => up to B-
- If you can finish Phase 2 with good programming styles + OO programming skills => up to B+
- If you can finish Phase 3 with good programming styles + OO programming skills => up to A-
- If you can finish Phase 4 with good programming styles + OO programming skills => up to A+

Each phase has various test cases (eg. Phase 1: 1_a.txt, 1_b.txt, etc..). If you get partial correct, your work is still considered. Eg. If you can pass 1a.txt only, you may get up to C-.

For "Good Programming Styles", note that proper indentations , code-layout formatting, proper , meaningful naming, well-designed classes, methods, fields are more important than writing comments.

Assignment Description

This is a simplified management system for a company to handle the grouping of employees and their annual leave.

An employee is given an amount of annual leaves (0 - 300 days) upon hired. How a normal employee take leave is handled very flexibly: no special approval is needed by his boss. For example, the following shows the output listing of the leaves of Bob and Carol:

```
> listLeaves
Bob:
2-Jan-2016 to 8-Jan-2016
Carol:
2-Jan-2016 to 8-Jan-2016
2-Mar-2016 to 9-Mar-2016
```

Each employee can join 1 or more teams, or can be no team. Each team has a team head. An employee could take multiple roles. As an extreme case, he could be the heads of 2 teams and a normal member of 3 teams. When we list the members of a team, the head is listed together, ordered by their names. For example, the following shows the listing of 2 teams:

```
> listTeamMembers
Production Team:
Bob (Head of Team)
Carol

Sales Team:
Ada
Bob (Head of Team)
Tim
```

Actually, being the head of a team has no significant privilege. Even worse, whenever he wants to take leave, he must find somebody to substitute him in the team as the *Acting Head* for that period. These acting head assignments could be seen when we list team members. Note that if one is the head of 2 or more teams, then one acting head needs to be assigned for each affected team. For example, below shows that Bob takes leaves in 2 periods, and assigns the acting heads for both the Production Team and the Sales Team.

```
> takeLeave|Bob|14-Jan-2016|18-Jan-2016|Production Team|Carol|Sales Team|Tim
Done.

> takeLeave|Bob|02-Jan-2016|04-Jan-2016|Production Team|Carol|Sales Team|Ada
Done.

> listTeamMembers
Production Team:
Bob (Head of Team)
Carol
Acting heads:
2-Jan-2016 to 4-Jan-2016: Carol
14-Jan-2016 to 18-Jan-2016: Carol

Sales Team:
Ada
Bob (Head of Team)
Tim
Acting heads:
2-Jan-2016 to 4-Jan-2016: Ada
14-Jan-2016 to 18-Jan-2016: Tim
```

Your task:

Implement this system based on your learning from Lab08, Lab09, and Lab10. Different test cases are given for each phase.

Phase 1 - Basic testing:	1_a.txt
Phase 1 - Undo/redo:	1_b.txt
Phase 1 - Exceptional cases:	1_c.txt
Phase 2 - Basic testing:	2_a.txt
Phase 2 - Undo/redo:	2_b.txt
Phase 2 - Exceptional cases:	2_c.txt
Phase 3 - Basic testing:	3_a1.txt, 3_a2.txt
Phase 3 - Undo/redo:	3_b1.txt, 3_b2.txt
Phase 3 - Exceptional cases:	3_c1.txt, 3_c2.txt
Phase 4 - Basic testing:	4_a1.txt, 4_a2.txt
Phase 4 - Undo/redo:	4_b1.txt, 4_b2.txt
Phase 4 - Exceptional cases:	4_c1.txt, 4_c2.txt

Below are the main requirements of each phase, and general guidelines. For further details of command formats and required outputs, refer to the styles in Lab08 - Lab10 and the given test cases for this assignment.

Phases

Phase 1 - Hiring employees, creating team, advancement of system date, listing of teams and employees.

Phase 2 - Leave taking by non-head and non-acting employees, listing of leaves.

Phase 3 - Adding members to teams, listing of roles and team members

Phase 4 - Leave taking by team heads: assignment of acting heads.

Note 1: Only the team member listing method needs to be revised to show the acting heads.

Note 2: Acting heads cannot take new leaves which overlap the acting periods.

General guidelines:

Sorting - For listing of teams, sort by the team names

For listing of employees, sort by the employee names

For listing of leaves of an employee, sort by the start day of the leave

For listing of team members, sort by the team names and then the employee names

For listing of roles of an employee, sort by the team names

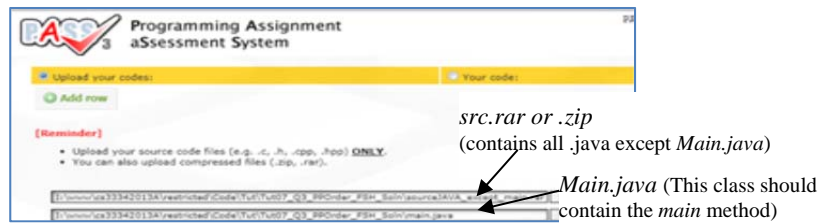
Name of source files -

You should name all command classes with the prefix: "Cmd", eg. "class CmdListLeaves", "class CmdTakeLeave"

Handling of errors -

You will need to add handling for many error cases. Most of them should be done by Exception Handling. You should name all Exception classes with prefix: "Ex", eg. "ExDateHasAlreadyPassed", "ExOverlappedLeaves"

Please submit them to PASS as shown below:



-- end --