Assignment - Team and Project Manpower Management

[Weight: 15 marks out of the final mark of this course]

Deadline: Apr 30, 2021 (Friday of revision week)

For late submissions, 2% of your original marks will be deducted if you hand in 1-day late (i.e. on May 01), 25% for 2-days (i.e. on May 02), assignments handed in on or after May 03 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,

Grading:

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

Various test cases are used for each phase (e.g. Phase 1: a1.txt, c3.txt, etc..). If you get partial correct, your work is still considered. E.g. If you can pass a1 – b3 only, your grade may be 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.
- During marking (May 03-10), selected students will be asked to meet me for discussion of your work.

Note:

Please apply what you learn from Lab08, Lab09 - Team Management and Lab10 Q1. You may reuse the code that you work for Lab08 – Lab10. Reusing these code would not be considered as plagiarism.

Please first finish your program for Lab09, then modify and add the required functionalities for this Assignment.

I. Introduction and Basic Requirements

This is a simplified management system for a company to handle the working teams, employees and project assignments. The company forms teams to work for projects. Each team has a leader when it is formed. Then other employees may be assigned to join the team as normal team members.

Two listings are shown below. The listing of teams shows the typical information of the teams: a team name, a team leader, the date when the team was setup, and a list of members. The listing of employees shows the teams that the employees belong to, if any.

```
> listTeams

Team Name      Leader    Setup Date   Members
All-rounder    Adam      1-Jan-2021   Andrew, Aron
Gas+Oil Gang   Gabriel   1-Jan-2021   Gavin, George
```

```
> listEmployees

Adam (All-rounder)
Andrew (All-rounder)
Aron (All-rounder)
Bill
Bob
Gabriel (Gas+Oil Gang)
Gavin (Gas+Oil Gang)
George (Gas+Oil Gang)
Peter
```

The listing of projects below shows the information of projects: a project has its project code, project title, project creation date, status, and the assigned team (with assignment date, completion date), if any.

```
> listProjects
Code     Project Title          Created on   Status       Assigned to   Assigned on  Completed on
P001     TSUEN WAN GAS          1-Feb-2021   Completed    All-rounder   2-Feb-2021   25-Mar-2021
P002     KWAI CHUNG CABLE       1-Feb-2021   Pending      --            --           --
P003     KWAI CHUNG SEWAGE      1-Feb-2021   Completed    SW Experts    18-Feb-2021  25-Mar-2021
P004     TAI WAI GAS            1-Mar-2021   In-progress  All-rounder   22-Mar-2021  --
```

For simplicity, we have the following assumptions and rules:

- All date entries are valid dates and are reasonable. E.g., the given project completion date is not earlier than the assignment date, which is not earlier than the project creation date.
- An employee cannot belong to two teams at the same time. That is, an employee can belong to one team only, either as the leader or a normal member.
- A project is assigned to at most one team only. But a team may be assigned to work on multiple projects at the same time. Assume that there is no personnel changes during the whole period. That is, at the beginning, teams and employees are created and employees join the teams. In your program, you do not need to handle the cases where employees change teams etc..
- Basically, when a project is created, the company gives a project code to the project. The first project has the code "P001", the second is "P002" etc.. (Hint: `String.format("P%03d", number_of_project)`). We assume that there are at most 999 projects. The project creation date is set to the system date, as illustrated below:

```
> createProject|TSUEN WAN GAS
Project created: [P001] TSUEN WAN GAS (1-Feb-2021)
```

- The starting status of the project is "Pending". Then the project is assigned to a team with a start day. The status of the project is "In-progress". When the team completes the project, the project status changes to "Completed"
- **State Pattern** should be used for project status (Lab06). There should be an interface in your source files named **PStatus.java**, and the implementing classes named **PStatus_Pending.java**, **PStatus_InProgress.java**, and **PStatus_Completed.java**.
  Inside **PStatus_InProgress.java** etc., you can have object fields that refer to the working Team etc.. (This is just one option. The final design is up to you.)

II. Advanced Requirements

In phase 2-3 of this assignment, we will have the following requirements:

Extension Projects

We allow the creation of **Extension Projects**, which are the projects as the follow-up of a previous project. To create an **Extension Project**, we need to provide the code of the parent project. The code of an Extension Project should be started with the code of the parent project, and appended with "-E" and then the numbering of the extension.

For example, two extension projects of "P003" are coded as "P003-E1", "P003-E2", as shown in the project listing below:

```
> listProjects
Code     Project Title          Created on   Status       Assigned to   Assigned on  Completed
on
P001     TSUEN WAN GAS          1-Feb-2021   In-progress  All-rounder   2-Feb-2021   --
P002     KWAI CHUNG CABLE       1-Feb-2021   Pending      --            --           --
P002-E1  Phase 2 for KC Cable   1-Mar-2021   Pending      --            --           --
P003     KWAI CHUNG SEWAGE      1-Feb-2021   In-progress  SW Experts    18-Feb-2021  --
P003-E1  KC Sewage Fixes 2021   18-Mar-2021  Pending      --            --           --
P003-E2  Phase 2 for KC Sewage  18-Mar-2021  Pending      --            --           --
```

For simplicity, we have the following assumptions and rules:

- Project extensions are not recursive. That is, there will be no extension project to extend another extension project.
- For simplicity of code generation, each project allows at most 9 extension projects.

External Support Staff

Each project can be assigned to one team, and an unlimited amount of support staff from other teams. For example, below is an example of assigning one team and 4 staff to work on project .

```
> assignProject|P003|SW Experts|Aron|Andrew|Bob|Bill
Done.

> listProjectStaff|P003
Project team: SW Experts
Project team members: Samuel (The Leader), Sawyer, Simon, Spencer, Steven
External support: Andrew, Aron, Bill, Bob
```

Suggestion of Team Assignment

Two projects could be related to each other if one is the extension project of another, or both are the extension projects of a parent project. When a project is pending for assignment, we may like to have suggestions of teams and employees to be assigned for the project. Below shows an example:

```
> giveAssignmentSuggestions|P003-E1

These teams have worked on related projects:
SW Experts: P003(Completed)
SW Experts B: P003-E2(In-progress)

These staff have worked on related projects:
Andrew: P003(Completed), P003-E2(In-progress)
Aron: P003(Completed)
Bill: P003(Completed), P003-E2(In-progress)
Bob: P003(Completed), P003-E2(In-progress)
Brian: P003-E2(In-progress)
Samuel: P003(Completed)
Sawyer: P003(Completed)
```

III. General Guidelines:

- The program needs to handle undo-redo of commands.

- Name of source files -
  You should name all command classes with the prefix: "`Cmd`", e.g. "`class CmdJoinTeam`", "`class CmdStartNewDay`"

- Most problem cases should be handled using Exception Handling as covered in this course.
  You should name all Exception classes with prefix: "Ex", e.g. "`ExEmployeeNameAlreadyExists`", "`ExProjectNotFound`"

- Please apply the following ordering of listing:

  Listing of employees : order by employee name
  Listing of teams : order by team name
  Listing of projects : order by project code

- Test cases -

  The given test cases and outputs are to show the functionalities that you need to implement. They also serve as specifications of input and output formats.

  However, note that they do not test rigorously for the accuracy of your program.
  For example, your program should be able to handle various sequences of undo-redo of different undoable commands appropriately. Also, a project may be assigned to a lot of support staff, instead of just up to four support staff in the given test cases.
  If your program fails to work appropriately, then your grade will be affected due to incorrect solution (despite that you might have obtained 100% correct in PASS).

- Concerns about efficiency: Assume that we have 1000 projects, 100 teams and 500 employees. For modern computers, keeping these records and providing sorting/searching are not a problem. Therefore, please spend more time and effort on modelling the entities involved in the case study.

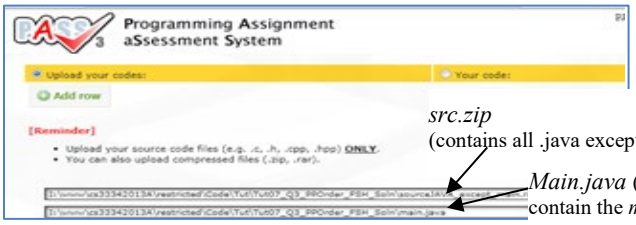
IV. Grading and requirements by phases:

The table below lists the main requirements and test cases of each phase. For command formats and required outputs, please refer to the styles in Lab08 Q2 to Lab10 Q1, and the contents in the given test cases and outputs at the course web.

Phase 1 (a) Hire employee and list employees (b) Setup team, employees join teams, list teams, and start new day (c) Create project and list projects	Basic testing:	a1.txt, b1.txt, c1.txt
	Undo/redo:	a2.txt, b2.txt, c2.txt
	Exceptional cases:	a3.txt, b3.txt, c3.txt
Phase 2 (d) Assign project to a team, mark completion of project, list projects by team (e) Create extension projects Note: • (d) You may revise the listing function for projects	Basic testing:	d1.txt, e1.txt
	Undo/redo:	d2.txt, e2.txt
	Exceptional cases:	d3.txt, e3.txt
Phase 3 (f) Assign project to a team and support staff, list project staff, list staff participations (g) Give assignment suggestions	Basic testing:	f1.txt, g1.txt
	Undo/redo:	f2.txt
	Exceptional cases:	f3.txt

V. Submission:

Please submit your work to PASS as shown below:



src.zip
(contains all .java except *Main.java*)

Main.java (This class should contain the *main* method)

-- end --