Problem 1. Design Thinking

Problem description

Among the following topics, choose **one** and design **two or more** classes to model the objects or concepts about it.

1. Books and bookshop (or bookshelf, library, etc.). The information of a book may include its title, authors, ISBN, price (if on sale in a bookshop), etc. A bookshop, bookshelf or library contains many books (as well as some other information), and people may add, remove, buy, borrow or return books. Design some classes that might be helpful in, for example, a bookshop management program.
2. A person's information includes his/her name, gender, address, email, phone number, etc. and an address consists of the country/region, city, street and some more details. We may need automatic filling of the information into some blanks (Google Chrome has this functionality), or we may want to generate a form.
3. The objects in Pacman: ghost, Pacman, food, game, level controller, etc.
4. Students and the courses they are going to select. A course has its credit, time, maximum number of attendees, dependencies, etc. A student may select several courses that don't conflict with each other.
5. Train tickets and passengers.
6. Job seekers and companies.
7. Posts in WeChat moments (朋友圈). A post has likes (点赞) and comments. A user has many posts, some of which are invisible due to certain settings.
8. Items for sale, shopping carts, and orders.

Note that this assignment is centered around **class designing**, which means that you don't have to fully implement all the functions since some of them are quite difficult. For those functions, you are free to write only a declaration and document its behavior, without actually implementing it (but make sure it is implementable).

Your classes should meet the following requirements:

1. Design and modeling: The design should make sense. What need to be stored as data members? What interfaces should be provided for users? What should be the behaviors of initialization and destruction?

2. Practice what you have learned:

   1. (10) All classes and members should be named in a meaningful and consistent manner.
   2. (10) Data members should be `private`, unless there is a strong reason for not to do so.
   3. (OJ) At least one class should have a user-provided constructor that is not a default, copy or move constructor.
   4. (10) Use constructor initializer lists whenever possible.
   5. (10) Use `const` whenever possible. In particular, pay attention to the `const`ness of member functions and the `const`ness in parameters.
   6. (OJ) Manual memory management, including `new` and `delete`, is not allowed. Make full use of standard library facilities like `std::string` and `std::vector`.

Grading

The grading of this problem consists of two parts: The OJ, and an offline check with TA.

- The OJ will check whether the requirements 2.3 and 2.6 are satisfied, and will also report errors on some typical bad practices.
- For the requirements 2.1, 2.2, 2.4 and 2.5, violation of any one will lead to 10 points deduction in your score.
- **Your code must be formatted!!** Violation of this will lead to **50 points** deduction in your score, and this time **we will not give you a chance to do it during the offline check**.

We will discuss the designs during the offline check, but take it easy -- a bad design will not lead to points deduction.