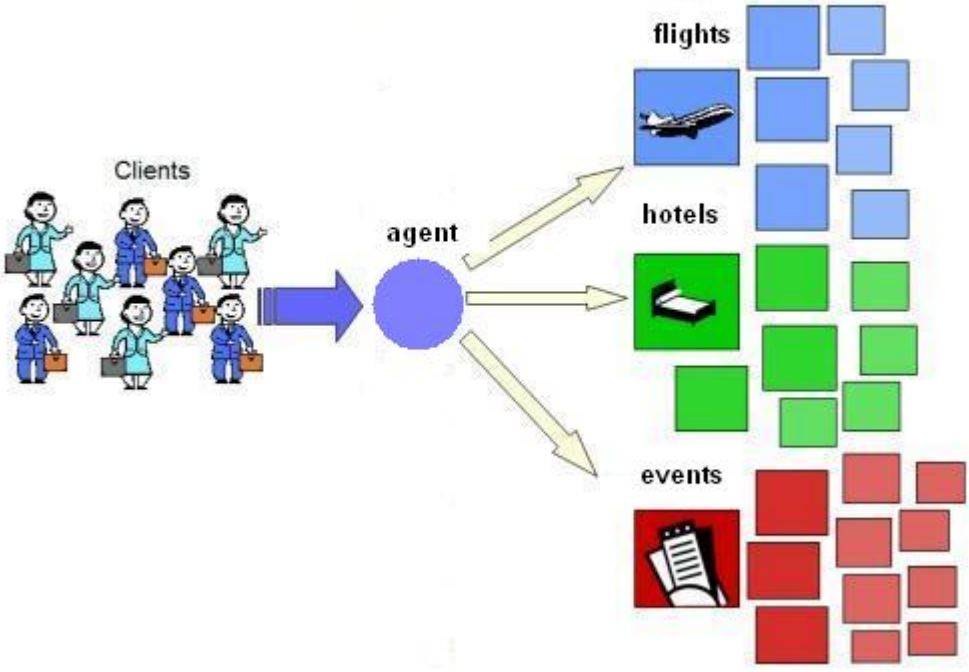
Assignment 2

Submit Deadline: 5:00pm 24 Oct 2018

1. Problem: Travel Agent / Smart Travel Agent

A travel agent offers a new product of selling travel packages for the 2020 Tokyo Olympic Games. Each package consists of three categories of goods: *flights*, *hotels* and *event tickets*.



The Olympic Games lasts ten days from day 0 to day 9. The travel agent has received a quota of hotel rooms and event tickets for these days.

Flights: The full price for one-way ticket between Sydney and Tokyo is $2000. However, a discount can be given depending on the days of arrival and departure.

The discount for a ticket from Sydney to Tokyo is *day*5%*. The discount for a ticket from Tokyo to Sydney is *(9-day)*5%.* For instance, if you travel to Tokyo on day 3 and return on day 8, the price of your tickets is:

2000*(1 – 3 * 5%) + 2000*(1- (9-8) * 5%) = 3600.

The idea here is "*the closer to the end, the cheaper for fly-in tickets; the closer to the end, the dearer for fly-out tickets"*. It is assumed that the supply of flight tickets is unlimited.

Hotels: There are three different types of hotel rooms in the Olympic Village: *3 star*, *4 star* and *5 star*. See the following table for the information about hotel rooms:

Types of rooms	**Full price (per room per night)**	**Total Rooms Available to the agent each day**	**Discount**
3 star hotels	$160	20	No discount
4 star hotels	$210	25	20% if vacancy > 50%
5 star hotels	$320	30	40% if vacancy > 50% 20% if vacancy > 20%

Hotel rooms are paid up-front. They are sold in vouchers. Each voucher is for a single day stay. For instance, if you travel to Tokyo on day 3 and return on day 8, you will need to buy five hotel vouchers, one for each day. You could be allocated to different types of rooms in different days.

Event tickets: There are totally 15 different events. See the following table for the detailed information:

No	**Event Date**	**Events**	**Ticket Price**	**Quantity available to the agent**
0	0	Opening	$2000	60
1	3	Soccer 1	$80	47
2	6	Soccer 2	$160	30
3	9	Soccer 3	$500	22
4	1	Track and Field 1	$80	50
5	2	Track and Field 2	$100	52
6	3	Track and Field 3	$120	42
7	4	Track and Field 4	$140	25
8	5	Swimming 1	$100	37
9	6	Swimming 2	$100	20
10	7	Gymnastics 1	$60	43
11	8	Gymnastics 2	$100	34
12	5	Basketball 1	$150	35
13	7	Basketball 2	$300	30
14	9	Closing	$800	40

Travel package: A travel package contains one fly-in air ticket (from Sydney to Tokyo), one fly-out ticket (from Tokyo to Sydney), the hotel rooms and a set of event tickets. A package is *feasible* if it satisfies the following conditions:

- The date of fly-in ticket is earlier than the date of fly-out ticket.
- The hotel rooms cover every night between the arrival and departure dates (not include the fly-out day).
- The hotel types should be the same or above the client's desired category.
- All the event tickets should be in the dates between fly-in and fly-out (can be in the same day of arrival or departure).
- Only the events a client requests can be included in the travel package to the client. They can be less but each package must contain at least one event.
- The total costs should not exceed client's budget.

Travel agent's profit: The travel agent charges 3% commission from air ticket and hotel voucher sales, and 10% from event ticket sales for each feasible package. There is no charge on unsuccessful requests.

You will be given a list of clients' requests (randomly generated). Each client's request consists of:

- the budget of the client (the upper limit the client can pay)
- desired hotel type (3-5 star)
- a set of desired events to attend (up to 10 events)

The requests are recorded in a text file. Download an example of request list from vUWS under this folder (requestList.txt). The format of each request in the text file is the following:

budget, *hotelType*, [$event_1$, $event_2$, …, $event_n$]

Note that there is no specification of travel dates in client's requests. By default, all travel package contains a return ticket, which cover all days for all the events and hotel stays. In other words, the days of events in the package are determined by the dates of travel and hotel stays.

2 Tasks and requirements

In this assignment, you are required to write a C++ program that implements a travel agent or a smart travel agent to design and sell travel packages to the potential clients. You will be given a list of client requests or randomly generate a list of client requests. Your travel agent should try to find a feasible travel package for each client based on the availability of hotel rooms, event tickets and client's budget. If you cannot find a feasible package for a client, indicate that the request is rejected. A smart travel agent should try to maximize its profit. You may translate your C++ into Java as an option of tasks but a C++ implementation is primary.

See following for more detailed requirements.

2.1 Pass level

Your program should include **at least** the following classes:

- a class called **ClientRequest** that specifies a single request of a client, containing the information of total budget, desired hotel types and a list of the events that the client wants to attend.
- a class called **Ticket** that specifies the common attributes of a ticket, including type of the ticket, valid date, full price and discounted price.
- three classes: **FlightTicket**, **HotelVoucher** and **EventTicket** that extend the **Ticket** class using inheritance.
- a class called **Package** that specifies a travel package, which contains two air-tickets, a set of hotel vouchers (one for each day) and a set of event tickets.
- a class called **TravelAgent** that takes a list of clients' requests, makes feasible travel package for all the clients if possible and report the outcome.

Requirement of implementation:

- Read the client requests from the provided text file and store the data in an array of **ClientRequest** objects.
- Output a report of all the feasible packages you produce and the list of rejected requests into a text file.

As a pass level program,

- You may ignore the information of price discount, the restrictions on hotel rooms and event tickets. Simply assume that all prices are in full and the supply of rooms and event tickets is unlimited.
- However, the total cost of each package should not exceed client's budget.

Note: You should implement all the classes required above with exactly the same names. However, you can create extra classes, or add more data items and member functions into above classes whenever needed no matter for the basic functionalities or any advanced functionalities.

2.2 Credit level

For students who seek credit or above, the calculation of packages will consider all aspects of a package, including price discount, quota of hotel rooms and event tickets. However, for credit only code, you do not have to optimize travel agent's profit.

2.3 Distinction level

Students who seek distinction or above are required to create a class called requestGenerator that simulates client requests. The class should implement a member function which is able to generate randomly 100-150 client requests.

Each request consists of:

- A randomly generated hotel category between 3 star to 5 star.
- Randomly generated 1-10 different events (at most one for each event).
- Randomly generated budget:
  - If the events include opening ceremony, randomly generate the budget between 4500+150*number-of-events to 7500+150*number-of-events.
  - If the events include closing ceremony, randomly generate the budget between 3800+150*number-of-events to 6800+150*number-of-events.
  - Otherwise, randomly generate the budget between 3250+150*number-of-events to 5250+150*number-of-events.

The format of each request should be:

budget, *hotelType*, [*event1*,*event2*, …, *event*n]

Download an example of this list from vUWS. Your program should be able to read this file.

2.3.1 High Distinction level

Students who seek for high distinction are requested to implement a smart travel agent which can maximize its profit. Note that clients' requests are not on the base of first-in-first-serve. The travel agent can choose who to offer the packages based on their budgets, desired hotel types, desired event tickets, availability of hotel rooms and event tickets to optimize its profit.

We will also check the quality of code for all the levels.

3. Submission

You need to submit your source code to vUWS for documentation purpose but you will gain your marks and feedback from your demonstration. The code should be purely written by yourself. No part of the code can be written by any other persons or copied from any other source. Submit the following declaration with your code (in a text file or world file).

DECLARATION

I hereby certify that no part of this assignment has been copied from any other student's work or from any other source. No part of the code has been written/produced for me by another person or copied from any other source.

I hold a copy of this assignment that I can produce if the original is lost or damaged.

Both the declaration and source code should be submitted via vUWS before the deadline. Your programs (.h, .cpp or .java) can be put in separate files (executable file is not required). All these files should be zipped into one file **with your student id as the zipped file name**. Submission that does not follow the format is not acceptable.

Email submissions are not acceptable.

5. Demonstration

You are required to demonstrate your program during **your scheduled** practical session in Week 13 on **25 or 26 October 2018**. **You will receive no marks if you fail the demonstration.** Note that it is your responsibility to get the appropriate compilers or IDEs to run your program. You are allowed to run your program from your laptop during demonstration. **The feedback to your work will be delivered orally during the demonstration.** No further feedback or comments are given afterward.

The program you demonstrate should be the same as the one you submit except that the comments in your program should be taken off before the demonstration.

You have to read this specification at least three times to get a full understanding of the problem.