AOOP Final Project – Program Elevator Fighter

1. Introduction

In the future, Engineering Building V is torn down and rebuilt because it is too old. The new building has 30 floors and there is only one elevator. Now, you are a professional engineer and asked to design the new elevator system. Firstly, you have to implement an elevator simulator to simulate the elevator system. The elevator is a special elevator, that is, it is a program elevator, which means that if one person wants to enter or leave the elevator, he/she has to solve a program. Each floor has its corresponding program. For example, you have to solve a prime problem on the first floor, solve a string problem on the second floor, and so on. In order to save everyone's time to take the elevator, you have two things to do. One is to improve the elevator scheduling, the other is to improve the programs performance on each floor. For finding out which systems designed in this class is the best one, we will have a competition.

2. Project Content and Requirement

In the future, Engineering Building V is torn down and rebuilt because it is too old. The new building has 30 floors and there is just only one elevator to conserve energy.

Now, you are a professional engineer asked to design the new elevator system **by using the techniques of the object-oriented software**. Your goal is to implement a working **smart elevator simulator program** that runs according to these specifications to achieve the following:

1. Improve the elevator scheduling and minimize the number of total moving floors to successfully meet the anticipated traffic requirements in the Building V.
2. Execute each floor program correctly and minimize the execution time of all the floor programs.

The elevator, which has a capacity of 10 persons, is designed to conserve energy, so it only moves when necessary. The elevator starts the day waiting with its door shut on floor 1 of the building. The elevator signals its arrival at a floor by turning on a light above the elevator door on that floor and by sounding a bell inside the elevator.

The elevator is a special elevator, that is, it is a program elevator, which means that if one person wants to leave the elevator, he/she has to solve a program. Each floor has its corresponding program. For example, you have to solve a prime problem on the first floor, solve a string problem on the second floor, and so on before leaving the elevator.

If there are "n" persons have the same destination floor f1, then you need to run the floor f1 program with n different data set, and each data set m times. The group with correct answer and less execution time will win this floor score (check answer first, and then compare the execution time).

These processes will be handled by the fair and smart "**Judge**". The "Judge" component of the elevator simulator provides one set of conditions:

1. the number of persons on each floor.
2. and their destination floors.
3. The floor number which the elevator is parked on.

When your elevator simulator accept these conditions:

1. The smart elevator simulator "creates" the persons for the specified floor, and places the person on that floor. The person's destination floor is never equal to the floor on which that the person arrives.
2. Update the status data of the "Elevator", and these status data will show on the windows GUI panel.
3. According to conditions provided by the "Judge", the "Scheduler" need to make the optimized decision:
   (a). Create the schedule to transport these persons with minimum number of total moving floors.
   (b). The Scheduler handshakes and communicates with "Judge" to complete its whole schedule.
   (c). Once these persons reach their destination floor:
      (I). They have to solve the corresponding floor program.
      (II). The "Judge" will monitor and determine the score of each floor for your team, and the corresponding run time messages will be displayed on windows GUI panel.
      (III). The "Judge" will monitor the whole process of executing floor programs and determine which team win the total floor scores.

The conclusion:

Your elevator simulator needs to

1. meet the anticipated traffic requirements,
2. solve each corresponding floor program within minimum time.
3. and optimize the traffic time.

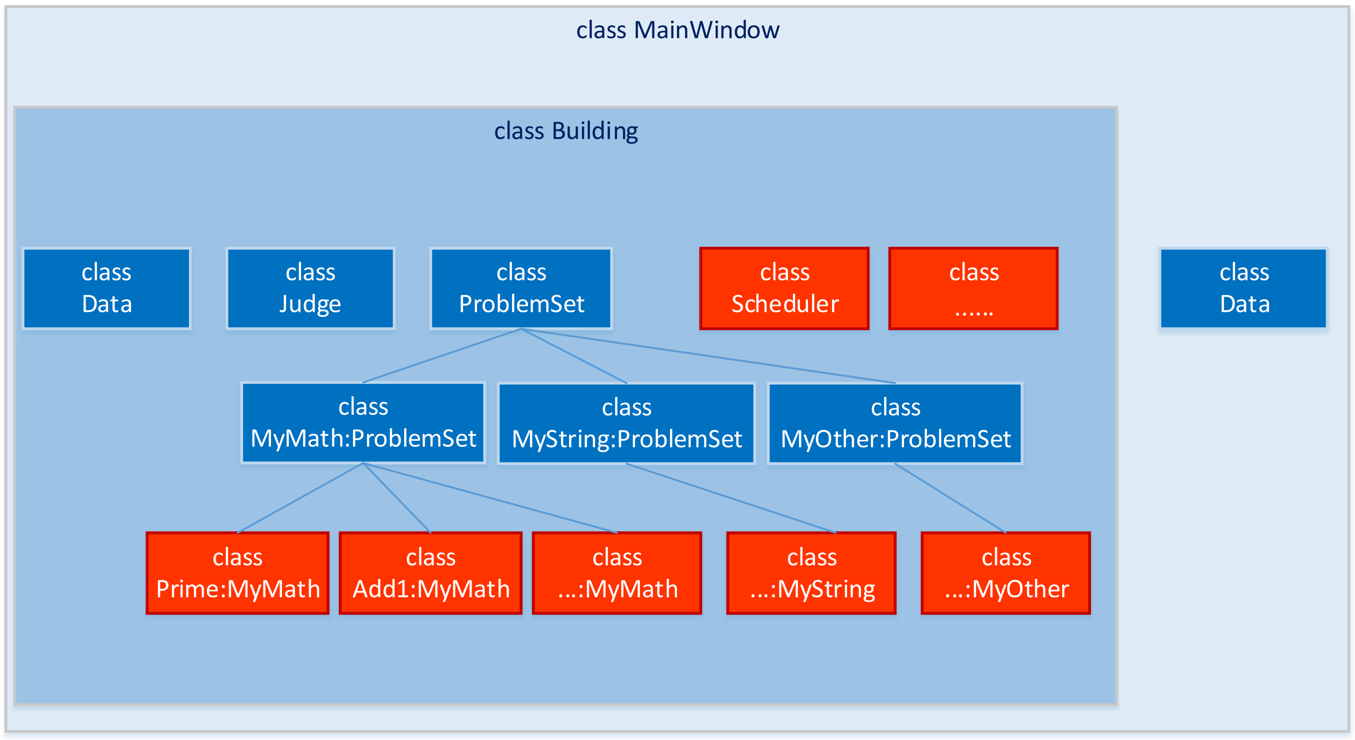
➢ Basic Requirement

In this elevator simulator program for the management part, you need to provide the following classes:

1. class Building,
2. class Scheduler
3. class Judge
4. class Floor
5. class Elevator
6. class Data.

According these specification, you need to add **five more classes** in your class hierarchy.

3. Overall System



4. Project Schedule

Week	Date	Content
8	2019.10/30.10/31	Project Introduction
10	2019.11/13.11/14	1. Introduction to UML Class Diagram 2. Announcement of Project System Specification 3. Announce Parts of Floor Programs 4. Floor Programs Using Polymorphism (Problem1) 5. Floor Programs (Wedx3, Thux3)
11	2019.11/20.11/21	1. Introduction to Database and MySQL (Part I) 2. Floor Program (Wedx2, Thux2)
12	2019.11/27.11/28	1. Introduction to Database and MySQL (Part II) 2. Class Diagram of Elevator Hierarchy Demo (One Page word) 3. Lobby Database Program (Problem1) 4. Floor Program (Wedx2, Thux2)
13	2019.12/04.12/05	1. Introduction to Database and MySQL (Part III) 2. Definition of Hand Shaking Flow (Input/Output of Scheduling/Floor Programs) 3. Elevator Scheduling Simulation (Problem1) 4. Floor Program (Wedx3, Thux3)
14	2019.12/11.12/12	1. Judge System Program – Part I 2. Elevator Scheduling Demo
15	2019.12/18.12/19	1. Definition of Competition Rules 2. Announcement of Double-Elimination Tournament Table 3. Homework05 (Problem1 and Problem2) – The demo time is the first half hour!!! 4. Judge System Program – Part II (Complete the system)
16	2019.12/25.12/26	Preliminary - Qualifying
19	**2020.01.16 (Thu) 13:00~**	**Final Competition** **Final Project Report (Deadline: 2020.01.16 (Thu) 23:55)**

5. Competition Rules

1) 兩人一組。
2) 比賽時採用的 Judge System 為助教提供，比賽當下會請各組將自己的檔案加進助教提供的 Project Files，因此請在比賽前熟悉好如何做這件事。
3) 初賽：初賽規制採 F1 排位賽制，最後一堂上機會讓各組在課堂上各自跑自己完成的專題程式，並可以重複將自己獨自跑分的結果利用程式傳到助教的主電腦，在該堂課結束前取最好的成績做排位，並依排位填入對應的雙淘汰制賽程表中。前 21 名做為種子選手可以直接進入決賽的第二輪。
4) 決賽：決賽採用雙淘汰制，依雙淘汰賽程表進行比賽，兩組同時在兩台電腦跑程式並比較積分結果，一場比 5 局，每一局輪流換電腦，第五局以丟銅板決定電腦。第 0 局先空跑程式得到第一組跑分結果，接下來 5 局再開始輪流互換比較計分。
5) 比賽時間 1 分鐘內沒到定位即喪失比賽資格，直接由另一組晉級。
6) 決賽每一場賽前會給各組 3 分鐘時間檢視對方的程式，若對對方程式有疑義可以向該場裁判助教提出，若經由助教判定程式有問題即判定失去資格。
7) 請不要作弊，請將心力放在增進程式效能，不要花心思想奧步。
8) 比賽時請不要賴皮。
9) 以和為貴。
10) 上述規則到比賽前皆會視狀況做合理修訂，若同學有任何疑問或覺得規則有所遺漏隨時皆可提出來給助教做合理修訂。

6. Program Rules

1) 不能在 Floor Program 任何地方預先建表或是做類似的事，包含 Class 任何地方、Constructor 裡等，所有的運算只能放在 solve()裡面(可以在 Run Time 時由程式自己建表)。
2) 可以在 Floor Program 定義其他的 function member，但是只能在 solve()裡面呼叫。
3) 每一樓的測資難度會有漸進性，由簡單到難。
4) 當程式當機或是 idle 太久(視窗畫面不更新>10s，由助教判定)，電梯內全部樓層的人全數死亡，並重新執行程式。
5) GUI 視窗上會有每層樓對應的 Checkbox 可以勾選是否放棄執行該題程式，若遇到執行到該樓層時會無法繼續執行下去最多 2 次，即應重新執行程式並將”放棄執行的 Checkbox”勾選讓程式可以繼續執行下去。

7. Competition Score

➢ **初賽**

總分數 = Elevator_Scheduling 分數(萬分制)*20% + Floor_Program 分數(萬分制)*80%

1) Elevator Scheduling (20%)
   - (a) 依據最終正確到達人數排序給本項積分，第一名 10000 分，依序遞減 200 分。
     上傳正確到達人數。
   - (b) 依據總移動樓層數排序給本項積分，第一名 10000 分，依序遞減 200 分。
     上傳總移動樓層數。
2) Floor Program (80%)
   - (a) 每一層樓測資共有 30 筆，因為測資難度有漸進性，因此每筆測資分數不一樣：
     第 1 筆 $(10000000000+2^{0})$ = 10000000001 分 (最簡單)
     第 2 筆 $(10000000000+2^{1})$ = 10000000002 分
     第 3 筆 $(10000000000+2^{2})$ = 10000000004 分
     ……
     第 19 筆 $(10000000000+2^{28})$ = 10268435456 分
     第 30 筆 $(10000000000+2^{29})$ = 10536870912 分 (最難)
     說明：對較多題分數會較高，若答對題數相同，答對較難的分數會較高。
   - (b) 各組將自己的每一輪的每題成績上傳後，每一題先依分數再依時間做排序，排序後照排名給該題的積分(第一名 100 點、第 2 名 99 點、……)，SQL 題因為不比時間，若分數相同則拿同樣積分(EX: 有 10 組分數最高且相同，則這 10 組都拿 100 點，第 11 名則拿 90 點)，總排名為每題積分累加後做排序，再依排序給本項積分。第一名 10000 分，依序遞減 200 分。

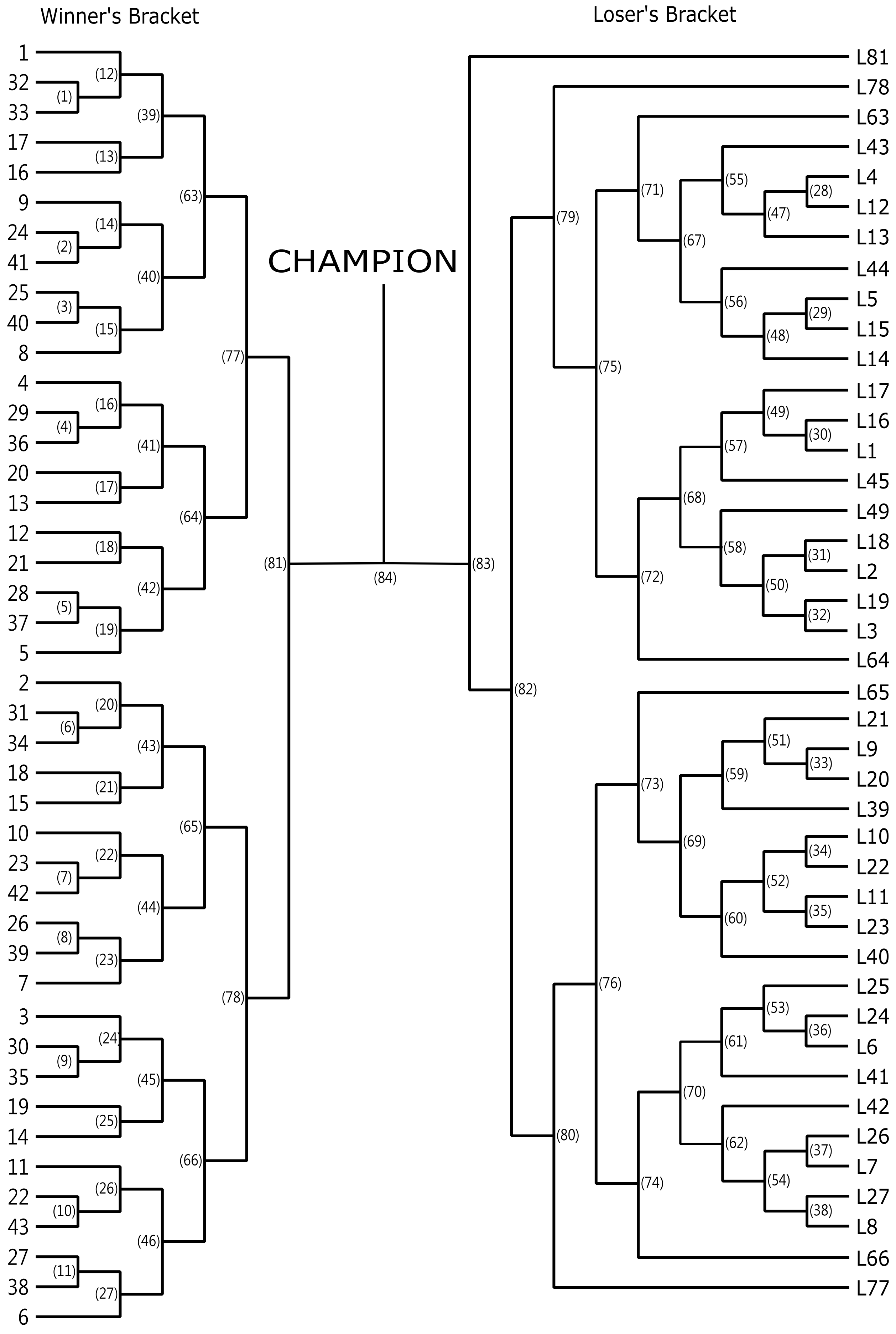
➢ **決賽**

總分數 = Elevator_Scheduling 分數(萬分制)*20% + Floor_Program 分數(萬分制)*80%
(比例可能會依據排位賽結果做適當調整)

1) Elevator Scheduling (20%)
   - (c) (最終正確到達人數/起始人數)*10000
   - (d) 總移動樓層數少的直接得到 10000
2) Floor Program (80%)
   - (a) 每一筆測資分數為 100 分。
   - (b) 一個人次(一筆測資)執行的 MyMath、MyString、 MyTree、MyOthers 的 Floor Program 會重複執行 N 次(N 依據該題難易度而定)。先比較該題答案，若只有一方答案正確則直接得到該題分數，若雙方答案皆正確則比較該筆測資執行時間，時間差異在 5%內雙方皆得到該題分數，若超過 5%則時間少的獲得該題分數。時間差異度算法定義如下:
     |A 組時間 - B 組時間|/AB 組時間較大者
   - (c) 一個人次(一筆測資)執行的 MyDatabase 的 Floor Program 只會執行 1 次，只要該題答案正確即可以得到分數，不比較時間。
   - (d) 本項分數 = 所得分數/(全部測資數*100) * 10000

3) Double-Elimination Tournament Table

種子位置：1 - 21



4) Project Score

1) Competition (65%)

依據比賽最終排名結果分配分數。

Rank	Score
Champion	100
(84) 共 1 組	98
(83) 共 1 組	96
(81) 共 1 組	94
(79)-(80) 共 2 組	90
(75)-(76) 共 2 組	87
(71)-(74) 共 4 組	83
(63)-(66) 共 4 組	80
(55)-(62) 共 8 組	76
(39)-(46) 共 8 組	72
(28)-(33) 總分數較高的 6 組	67
(28)-(33) 總分數較低的 5 組	63
共 43 組	AVG: 76.63 STD: 9.85

2) Report (35%)

5) Report

Deadline: 2020/01/16(Thu) 23:55

Your report should include the following topics.

A. Team Members

(50%) 0710987 - 蔡國語

(50%) 0710078 - 韓英文

請務必附上分工百分比(請自行協調列出每個人的分工比例)，沒寫即以 50%和 50%計。

Example: 若這個專題成績總分為 80 分，組員 A 和 B 分工比例為 50%和 50%，則兩人專題總成績就都為 80 分，若是 A 和 B 分工比例為 60%和 40%，則每差 10%分數會差 5 分，因此 A 分數為 80 分，B 分數為 70 分。(請以 10%為基本單位)

B. Class Hierarchy

Please draw the class hierarchy using UML which at least includes data members and member functions.

C. Scheduling Algorithm

Please illustrate your scheduling algorithm and prove the performance of this algorithm. You can use block diagram and text description to illustrate it.

D. Floor Program Algorithms

Please illustrate your algorithms of each floor program and prove the performance of these algorithms. You can use block diagram and text description to illustrate it.

E. Question and Discussion

Discuss the problems you meet in this project and explain how to solve it.

F. Project Result

Illustrate and explain your project result. Maybe you can show some pictures.

G. References

Please list your references.

H. 心得感想

就是心得與感想。