Final Project – Group 8

- Create a Java project named **Final project**
- Read instructions and create classes needed. You are supposed to add 11 classes (**Course, Department, ACCT, BA, FIN, IB, MAB, MIS, RMI, STAT and *Main***) to the project.
- Download required and partially required subjects csv files of eight departments.
- All instance variables are private or protected. Please use public interfaces to access private variables.

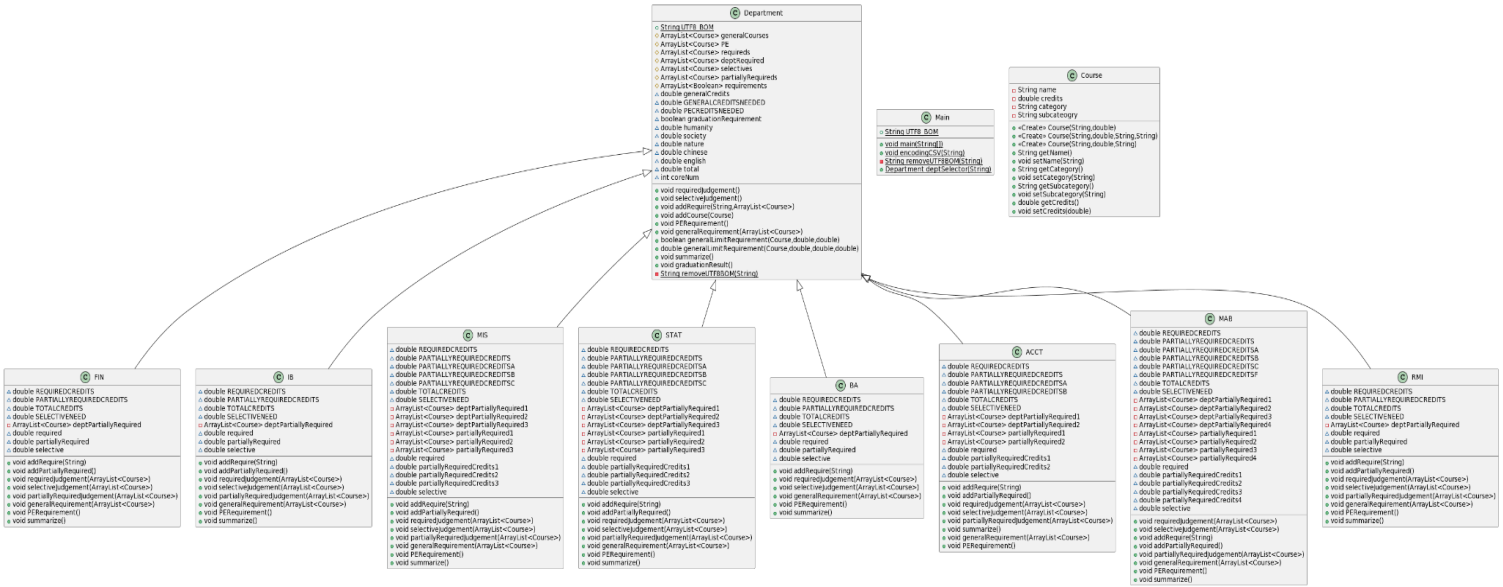
Following figure shows the inheritance, attributes and methods for each class. For setter and getter please refer to Figure 1.

Description:

The College of Commerce in NCCU has eight departments. We want to help students calculate whether they meet the credits requirements for graduation. Each *course* has its course name, credits, category and subcategory. We use csv files to store required subjects and partially required subjects of eight different departments. And eight department classes (*ACCT, BA and so on*) have similar attributes inherited from the original department class, but have some differences in requirements. Please check the following class diagram and implement the result.
At the end, you have to show the course-taking information of this student.

Class Diagram:

- github link



1. Create **Course** Class

Course

Modifier and type	Method (or Variable) and description
Instance variable	
String	name The name of the course
double	credits The credits of the course
String	category The category of the course
String	subcategory The subcategory of the course
Constructor	
Course(String name, double credits) Enable to instantiate the object of Course with a given name and credits.	
Course(String name, double credits,String category,String subcategory) Enable to instantiate the object of Course with a given name, credits, category and subcategory.	
Course(String name, double credits,String category) Enable to instantiate the object of Course with a given name, credits and category	
Methods	
-	4 getters and 4 setters for name, category, subcategory, credits

2. Create **Department** Class

Department	
Modifier and type	**Method (or Variable) and description**
variable	
final String	(static) **UTF8_BOM** declare “\uFEFF” as a final String for the encoding purpose in Chinese.
ArrayList<Course>	**generalCourses** Store general courses that the student took
ArrayList<Course>	**PE** Store PE courses that the student took
ArrayList<Course>	**requireds** Store required courses that the student took

ArrayList<Course>	**deptRequired** Store the required courses that the student needs for graduation
ArrayList<Course>	**selectives** Store the selective courses that the student took
ArrayList<Course>	**partiallyRequireds** Store partially required courses that the student took
ArrayList<Boolean>	**requirements** Use a Boolean mark ArrayList to check whether a student meets the graduation requirement of credits under each category.
double	**generalCredits** Store general course credits the student has currently.
final double	**GENERALCREDITSNEEDED** Store the general course credits that the student needs for graduation.
final double	**PECREDITSNEEDED** Store the number of PE courses that the student has to take for graduation.
boolean	**graduationRequirement** Declare a boolean mark to judge whether the student meets the requirement of graduation.
Constructor	
-	
Methods	
void	**requiredJudgement()** Declare an empty method for the usage of inheritance.
void	**selectiveJudgement()** Declare an empty method for the usage of inheritance.
void	**addRequire(String fileName,ArrayList<Course> courses)** Add required courses list of certain category (required courses or partially required course) to the Department class's instance variable which is used to store the requirement of certain category (e.g. deptRequired)
void	**addCourse(Course course)** Add courses that the user has taken to the ArrayList of each category.
void	**PERequirement()**

	Test whether the student took enough PE credits.
void	**generalRequirement(ArrayList<Course> generalCourses)** Test whether the student meets the generalRequirement thresholds of each category. (use two Boolean ArrayLists to store the results and print out the credits information of each general requirement category.)
boolean	**generalLimitRequirement(Course course, double min, double max)** Return boolean value of "false" if requirement credits are lower than minimum limit, if not, return "true".
double	**generalLimitRequirement(Course course, double credits, double min, double max)** Return the number of requirement credits (if it is higher than maximum, return maximum credits).
void	**summarize()** use the for-each loop to store the results of pass or not pass of requirements (pass:true / not pass:false).
void	**graduationResult()** Print the result of whether the student took enough credits for graduation, and show the detailed credits information for each category of PE and general requirements (including each category).
static String	**removeUTF8BOM(String s)** return a String without the prefix of BOM(Byte of Order Mark) to make sure data can be read properly.

3. Create **ACCT** Class

ACCT	
Modifier and type	**Method (or Variable) and description**
Instance variable	
final double	REQUIREDCREDITS The required credits that are needed
final double	PARTIALLYREQUIREDCREDITS All the partially required credits that are needed
final double	PARTIALLYREQUIREDCREDITSA The partially required credits of category A that are needed
final double	PARTIALLYREQUIREDCREDITSB The partially required credits of category B that are needed

final double	TOTALCREDITS Total credits that are needed to graduate
final double	SELECTIVENEED The selectives credits that are needed
ArrayList<Course>	deptPartiallyRequired1 The ArrayList to store the courses of the department partially required of category A that are needed
ArrayList<Course>	deptPartiallyRequired2 The ArrayList to store the courses of the department partially required of category B that are needed
ArrayList<Course>	partiallyRequired1 The ArrayList to store the courses of the department partially required of category A that user has taken
ArrayList<Course>	partiallyRequired2 The ArrayList to store the courses of the department partially required of category B that user has taken
double	required The required credits that user has taken
double	partiallyRequiredCredits1 The partially required credits of category A that user has taken
double	partiallyRequiredCredits2 The partially required credits of category B that user has taken
double	selective The selective credits that user has taken
Constructor	
-	
Methods	
void	**addRequire(String fileName)** Use the super method to add requiredCredits threshold information to ACCT.
void	**addPartiallyRequired()** Use the super method to add partiallyRequiredCredits threshold information to ACCT.
void	**requiredJudgement(ArrayList<Course> courses)** Test the "required" courses and credits between users taken and the threshold.

void	**selectiveJudgement(ArrayList<Course> courses)** Test whether the student meets the threshold of selective credits and print out the credits information of selective needs.
void	**partiallyRequiredJudgement(ArrayList<Course> courses)** Store course-taken information to the subcategories of different ArrayLists and test whether the student meets the threshold of PartiallyRequired credits. Finally, print out the credits information of partiallyRequired needs.
void	**summarize()** Override the super summarize method of the class Department.
void	**generalRequirement(ArrayList<Course> generalCourses)** Override the super generalRequirement method of the class Department.
void	**PERequirement()** Override the super PERequirement method of the class Department.

4. Create **BA** Class

BA	
Modifier and type	**Method (or Variable) and description**
Instance variable	
final double	REQUIREDCREDITS The required credits that are needed
final double	PARTIALLYREQUIREDCREDITS All the partially required credits that are needed
final double	TOTALCREDITS Total credits that are needed to graduate
final double	SELECTIVENEED The selectives credits that are needed
ArrayList<Course>	deptPartiallyRequired The ArrayList to store the courses of the department partially required that are needed
double	required The required credits that user has taken
double	partiallyRequired The partially required credits that user has taken
double	selective

	The selective credits that user has taken
Constructor	
-	
Methods	
void	**addRequire(String fileName)** Use the super method to add requiredCredits threshold information to BA.
void	**requiredJudgement(ArrayList<Course> courses)** Test the "required" courses and credits between users taken and the threshold.
void	**selectiveJudgement(ArrayList<Course> courses)** Test whether the student meets the threshold of selective credits and print out the credits information of selective needs.
void	**summarize()** Override the super summarize method of the class Department.
void	**generalRequirement(ArrayList<Course> generalCourses)** Override the super generalRequirement method of the class Department.
void	**PERequirement()** Override the super PERequirement method of the class Department.

5. Create **FIN** Class

FIN	
Modifier and type	**Method (or Variable) and description**
Instance variable	
final double	REQUIREDCREDITS The required credits that are needed
final double	PARTIALLYREQUIREDCREDITS All the partially required credits that are needed
final double	TOTALCREDITS Total credits that are needed to graduate
final double	SELECTIVENEED The selectives credits that are needed
ArrayList<Course>	deptPartiallyRequired

	The ArrayList to store the courses of the department partially required that are needed
double	required The required credits that user has taken
double	partiallyRequired The partially required credits that user has taken
double	selective The selective credits that user has taken
Constructor	
-	
Methods	
void	**addRequire(String fileName)** Use the super method to add requiredCredits threshold information to FIN.
void	**addPartiallyRequired()** Use the super method to add partiallyRequiredCredits threshold information to FIN.
void	**requiredJudgement(ArrayList<Course> courses)** Test the "required" courses and credits between users taken and the threshold.
void	**selectiveJudgement(ArrayList<Course> courses)** Test whether the student meets the threshold of selective credits and print out the credits information of selective needs.
void	**partiallyRequiredJudgement(ArrayList<Course> courses)** Store course-taken information to the subcategories of different ArrayLists and test whether the student meets the threshold of PartiallyRequired credits. Finally, print out the credits information of partiallyRequired needs.
void	**summarize()** Override the super summarize method of the class Department.
void	**generalRequirement(ArrayList<Course> generalCourses)** Override the super generalRequirement method of the class Department.
void	**PERequirement()** Override the super PERequirement method of the class Department.

6. Create **IB** Class

IB	
Modifier and type	**Method (or Variable) and description**
Instance variable	
final double	REQUIREDCREDITS The required credits that are needed
final double	PARTIALLYREQUIREDCREDITS All the partially required credits that are needed
final double	TOTALCREDITS Total credits that are needed to graduate
final double	SELECTIVENEED The selectives credits that are needed
ArrayList<Course>	deptPartiallyRequired The ArrayList to store the courses of the department partially required that are needed
double	required The required credits that user has taken
double	partiallyRequired The partially required credits that user has taken
double	selective The selective credits that user has taken
Constructor	
-	
Methods	
void	**addRequire(String fileName)** Use the super method to add requiredCredits threshold information to IB.
void	**addPartiallyRequired()** Use the super method to add partiallyRequiredCredits threshold information to IB.
void	**requiredJudgement(ArrayList<Course> courses)** Test the "required" courses and credits between users taken and the threshold.
void	**selectiveJudgement(ArrayList<Course> courses)** Test whether the student meets the threshold of selective credits and print out the credits information of selective needs.

void	**partiallyRequiredJudgement(ArrayList<Course> courses)** Store course-taken information to the subcategories of different ArrayLists and test whether the student meets the threshold of PartiallyRequired credits. Finally, print out the credits information of partiallyRequired needs.
void	**summarize()** Override the super summarize method of the class Department.
void	**generalRequirement(ArrayList<Course> generalCourses)** Override the super generalRequirement method of the class Department.
void	**PERequirement()** Override the super PERequirement method of the class Department.

7. Create **MAB** Class

MAB	
Modifier and type	**Method (or Variable) and description**
Instance variable	
final double	REQUIREDCREDITS The required credits that are needed
final double	PARTIALLYREQUIREDCREDITS All the partially required credits that are needed
final double	PARTIALLYREQUIREDCREDITSA The partially required credits of category A that are needed
final double	PARTIALLYREQUIREDCREDITSB The partially required credits of category B that are needed
final double	PARTIALLYREQUIREDCREDITSC The partially required credits of category C that are needed
final double	PARTIALLYREQUIREDCREDITSF The partially required credits of category F that are needed
final double	TOTALCREDITS Total credits that are needed to graduate
final double	SELECTIVENEED The selectives credits that are needed
ArrayList<Course>	deptPartiallyRequired1 The ArrayList to store the courses of the department partially required of category A that are needed

ArrayList<Course>	deptPartiallyRequired2 The ArrayList to store the courses of the department partially required of category B that are needed
ArrayList<Course>	deptPartiallyRequired3 The ArrayList to store the courses of the department partially required of category C that are needed
ArrayList<Course>	deptPartiallyRequired4 The ArrayList to store the courses of the department partially required of category F that are needed
ArrayList<Course>	partiallyRequired1 The ArrayList to store the courses of the department partially required of category A that user has taken
ArrayList<Course>	partiallyRequired2 The ArrayList to store the courses of the department partially required of category B that user has taken
ArrayList<Course>	partiallyRequired3 The ArrayList to store the courses of the department partially required of category C that user has taken
ArrayList<Course>	partiallyRequired4 The ArrayList to store the courses of the department partially required of category F that user has taken
double	required The required credits that user has taken
double	partiallyRequiredCredits1 The partially required credits of category A that user has taken
double	partiallyRequiredCredits2 The partially required credits of category B that user has taken
double	partiallyRequiredCredits3 The partially required credits of category C that user has taken
double	partiallyRequiredCredits4 The partially required credits of category F that user has taken
double	selective The selective credits that user has taken
Constructor	
-	
Methods	

void	**addRequire(String fileName)** Use the super method to add requiredCredits threshold information to MAB.
void	**requiredJudgement(ArrayList<Course> courses)** Test the "required" courses and credits between users taken and the threshold.
void	**selectiveJudgement(ArrayList<Course> courses)** Test whether the student meets the threshold of selective credits and print out the credits information of selective needs.
void	**partiallyRequiredJudgement(ArrayList<Course> courses)** Store course-taken information to the subcategories of different ArrayLists and test whether the student meets the threshold of PartiallyRequired credits. Finally, print out the credits information of partiallyRequired needs.
void	**summarize()** Override the super summarize method of the class Department.
void	**generalRequirement(ArrayList<Course> generalCourses)** Override the super generalRequirement method of the class Department.
void	**PERequirement()** Override the super PERequirement method of the class Department.

8. Create **MIS** Class

MIS	
Modifier and type	**Method (or Variable) and description**
Instance variable	
final double	REQUIREDCREDITS The required credits that are needed
final double	PARTIALLYREQUIREDCREDITS All the partially required credits that are needed
final double	PARTIALLYREQUIREDCREDITSA The partially required credits of category A that are needed
final double	PARTIALLYREQUIREDCREDITSB The partially required credits of category B that are needed
final double	PARTIALLYREQUIREDCREDITSC The partially required credits of category C that are needed

final double	TOTALCREDITS Total credits that are needed to graduate
final double	SELECTIVENEED The selectives credits that are needed
ArrayList<Course>	deptPartiallyRequired1 The ArrayList to store the courses of the department partially required of category A that are needed
ArrayList<Course>	deptPartiallyRequired2 The ArrayList to store the courses of the department partially required of category B that are needed
ArrayList<Course>	deptPartiallyRequired3 The ArrayList to store the courses of the department partially required of category C that are needed
ArrayList<Course>	partiallyRequired1 The ArrayList to store the courses of the department partially required of category A that user has taken
ArrayList<Course>	partiallyRequired2 The ArrayList to store the courses of the department partially required of category B that user has taken
ArrayList<Course>	partiallyRequired3 The ArrayList to store the courses of the department partially required of category C that user has taken
double	required The required credits that user has taken
double	partiallyRequiredCredits1 The partially required credits of category A that user has taken
double	partiallyRequiredCredits2 The partially required credits of category B that user has taken
double	partiallyRequiredCredits3 The partially required credits of category C that user has taken
double	selective The selective credits that user has taken
Constructor	
-	
Methods	
void	**addRequire(String fileName)**

	Use the super method to add requiredCredits threshold information to MIS.
void	**addPartiallyRequired()** Use the super method to add partiallyRequiredCredits threshold information to MIS.
void	**requiredJudgement(ArrayList<Course> courses)** Test the "required" courses and credits between users taken and the threshold.
void	**selectiveJudgement(ArrayList<Course> courses)** Test whether the student meets the threshold of selective credits and print out the credits information of selective needs.
void	**partiallyRequiredJudgement(ArrayList<Course> courses)** Store course-taken information to the subcategories of different ArrayLists and test whether the student meets the threshold of PartiallyRequired credits and print out the credits information of partiallyRequired needs.
void	**summarize()** Override the super summarize method of the class Department.
void	**generalRequirement(ArrayList<Course> generalCourses)** Override the super generalRequirement method of the class Department.
void	**PERequirement()** Override the super PERequirement method of the class Department.

9. Create **RMI** Class

RMI	
Modifier and type	**Method (or Variable) and description**
Instance variable	
final double	REQUIREDCREDITS The required credits that are needed
final double	PARTIALLYREQUIREDCREDITS All the partially required credits that are needed
final double	TOTALCREDITS Total credits that are needed to graduate
final double	SELECTIVENEED The selectives credits that are needed

ArrayList<Course>	deptPartiallyRequired The ArrayList to store the courses of the department partially required that are needed
ArrayList<Course>	partiallyRequired The partially required credits that user has taken
double	required The required credits that user has taken
double	partiallyRequired The partially required credits that user has taken
double	selective The selective credits that user has taken
Constructor	
-	
Methods	
void	**addRequire(String fileName)** Use the super method to add requiredCredits threshold information to RMI.
void	**addPartiallyRequired()** Use the super method to add partiallyRequiredCredits threshold information to RMI.
void	**requiredJudgement(ArrayList<Course> courses)** Test the "required" courses and credits between users taken and the threshold.
void	**selectiveJudgement(ArrayList<Course> courses)** Test whether the student meets the threshold of selective credits and print out the credits information of selective needs.
void	**partiallyRequiredJudgement(ArrayList<Course> courses)** Store course-taken information to the subcategories of different ArrayLists and test whether the student meets the threshold of PartiallyRequired credits. Finally, print out the credits information of partiallyRequired needs.
void	**summarize()** Override the super summarize method of the class Department.
void	**generalRequirement(ArrayList<Course> generalCourses)** Override the super generalRequirement method of the class Department.
void	**PERequirement()**

	Override the super PERequirement method of the class Department.

10. Create **STAT** Class

STAT	
Modifier and type	**Method (or Variable) and description**
Instance variable	
final double	REQUIREDCREDITS The required credits that are needed
final double	PARTIALLYREQUIREDCREDITS All the partially required credits that are needed
final double	PARTIALLYREQUIREDCREDITSA The partially required credits of category A that are needed
final double	PARTIALLYREQUIREDCREDITSB The partially required credits of category B that are needed
final double	PARTIALLYREQUIREDCREDITSC The partially required credits of category C that are needed
final double	TOTALCREDITS Total credits that are needed to graduate
final double	SELECTIVENEED The selectives credits that are needed
ArrayList<Course>	deptPartiallyRequired1 The ArrayList to store the courses of the department partially required of category A that are needed
ArrayList<Course>	deptPartiallyRequired2 The ArrayList to store the courses of the department partially required of category B that are needed
ArrayList<Course>	deptPartiallyRequired3 The ArrayList to store the courses of the department partially required of category C that are needed
ArrayList<Course>	partiallyRequired1 The ArrayList to store the courses of the department partially required of category A that user has taken
ArrayList<Course>	partiallyRequired2 The ArrayList to store the courses of the department partially required of category B that user has taken

ArrayList<Course>	partiallyRequired3 The ArrayList to store the courses of the department partially required of category C that user has taken
double	required The required credits that user has taken
double	partiallyRequiredCredits1 The partially required credits of category A that user has taken
double	partiallyRequiredCredits2 The partially required credits of category B that user has taken
double	partiallyRequiredCredits3 The partially required credits of category C that user has taken
double	selective The selective credits that user has taken
Constructor	
-	
Methods	
void	**addRequire(String fileName)** Use the super method to add requiredCredits threshold information to STAT.
void	**addPartiallyRequired()** Use the super method to add partiallyRequiredCredits threshold information to STAT.
void	**requiredJudgement(ArrayList<Course> courses)** Test the "required" courses and credits between users taken and the threshold.
void	**selectiveJudgement(ArrayList<Course> courses)** Test whether the student meets the threshold of selective credits and print out the credits information of selective needs.
void	**partiallyRequiredJudgement(ArrayList<Course> courses)** Store course-taken information to the subcategories of different ArrayLists and test whether the student meets the threshold of PartiallyRequired credits. Finally, print out the credits information of partiallyRequired needs.
void	**summarize()** Override the super summarize method of the class Department.
void	**generalRequirement(ArrayList<Course> generalCourses)** Override the super generalRequirement method of the class

	Department.
void	**PERequirement()** Override the super PERequirement method of the class Department.

11. Create ***Main*** Class

Main	
Modifier and type	**Method (or Variable) and description**
Instance variable	
final String	(static) **UTF8_BOM** declare "\uFEFF" as a final String for the encoding purpose in Chinese.
Methods	
static void	**encodingCSV**(String file) This method works as the description below : (1) read the file of the user input (2) create a Department object and cast it to the specific department object according to the file user provided (the format of the file is provided below) (3) call the summary( ) method of user's department object to print the result
static Department	**deptSelector(String dept)** To determine the specific department of user input and cast it to the Department obj created under encodingCSV( ) method ; and add the requirement course list (e.g.YourDept_Required.csv) according to the specific department of user input
static String	**removeUTF8BOM(String s)** return a String without the prefix of BOM(Byte of Order Mark) to make sure data can be read properly.

Input File Format Example:

1. **User file :**
   * If the course of a certain category which doesn't have a subcategory please enter "none" as a subcategory.

```
系所 IB,,,
初級會計學（一）,3,必修,none
社會責任與倫理,1,必修,none
初級會計學（二）,3,必修,none
經濟學,3,必修,none
國際財務管理,3,群修,none
國際經貿法專題,3,群修,none
物理學史,3,通識,自然核通
人文學史,3,通識,人文核通
社會學的想像,3,通識,社會核通
人文地理學,10,通識,社會
散文賞析,4,通識,中文
大學英文,6,通識,英文
體育A,0,體育,none
體育B,0,體育,none
體育C,0,體育,none
體育D,0,體育,none
資訊概論,3,體育,自然
```

2. **Requirement Course List File :**

```
社會責任與倫理,1
微積分,3
經濟學,6
民法概要,2
初級會計學（一）,3
初級會計學（二）,3
成本管理會計（一）,3
統計學,6
中級會計學（一）,3
商事法,2
成本管理會計（二）,3
中級會計學（二）,3
審計學（一）,3
中級會計學（三）,3
高級會計學（一）,3
會計資訊系統（一）,3
高級會計學（二）,3
審計學（二）,3
會計資訊系統（二）,2
稅務法規,3
財務報表分析,3
電腦審計,1
```

Output Example:

```
<必修課程>

應修 48.00 學分
已修 13.00 學分
尚缺 35.00 學分
需要補齊的必修課程：
初級會計學（二） 微積分 統計學 管理學 財務管理 行銷管理 社會責任與倫理 國際經貿法 國貿理論與政策 國際金融

-----------------------------------------------------------------------------------------------------
<選修課程>

應修 43.00 學分
尚缺 43.00 學分

-----------------------------------------------------------------------------------------------------
<群修課程>

已修 6.00 學分
尚缺 3.00 學分
請挑以下其中 1 堂課修習：
國際經貿法專題 國際行銷管理 貿易與投資專題 國際金融專題

-----------------------------------------------------------------------------------------------------
<通識及體育課程>

已修 25.00 學分
尚缺 3.00 學分

-----------------------------------------------------------------------------------------------------
<通識及體育課程細項>

體育已修 5 門
已通過體育課修習標準

國文通識（3~6）
已修 4.00 學分
已過下限 還能修 2.00 學分
```

Our code: