**Final Project**

**Objective:**

Enable students to easily confirm their current earned credits, and view required general education and physical education courses upon entering the course selection system.

**Situation:**

While the existing course inquiry system at National Chengchi University provides features such as subject name search and tracking, students still need to go through the process of confirming required courses and searching for general education and physical education courses, making the operation cumbersome and inconvenient. Therefore, we have decided to optimize the system. Firstly, the system will recognize the user's department and provide three main course selection options: required courses, general education, and physical education. When the user chooses to view required courses, the system will list all required courses that match the user's department. General education and physical education will also display corresponding course lists. In the end, users can add the courses of interest to their tracking list through this customized system. The system not only presents the user's tracking list but also assists in calculating the required credits for graduation, helping students plan and arrange course selections more conveniently. Our goal is to enhance the user experience through customization, making course selection easier and more efficient for students!

**分工表：**

**According to the information below, create a java class:**

1. Create **Course** class

| **Course** | |
| --- | --- |
| **Modifier and type** | **Method (or Variable) and description** |
| **Instance variable** | |
| **int** | credit  The course credit of this course. |
| **int** | id  The course number of this course. |
| **String** | teacher  The teacher’s name. |
| **String** | name  The course name of this course. |
| **String** | time  The time of this course. |
| **boolean** | empty  Indicating whether the course is empty, used for error handling or special situations. |
| **boolean** | selected  Indicating whether the course has been selected. |
| **boolean** | core  Indicating whether the course is a core course. |
| **Constructor** | |
| **Course(String name, int credit, int id, String time, String teacher, boolean core)**  Enable to construct a course object with given course name, credit, id, time, teacher, boolean core.  **Course(boolean empty)**  Enable to construct a course object marked as "empty". | |
| **Instance Methods** | |
| - | 5 getter for 5 attributes (getCredit(),getID(),getName(),getTime(),getTeacher()). |
| **boolean** | checkEmpty()  Check if the course is empty. |
| **boolean** | checkSelected()  Check if the course has been selected. |
| **void** | setSelected(boolean selected)  Set the course to a selected state. |
| **boolean** | checkCore()  Check if the course is a core course. |
| **String** | toString()  Return a String description for the course information.  (See output below) |

2. Create **NCCUStudent** class

| **NCCUStudent** | |
| --- | --- |
| **Modifier and type** | **Method (or Variable) and description** |
| **Instance variable** | |
| int | stdID  The student ID. |
| int | grade  The student’s grade. |
| int | department  The student’s department. |
| int | sportCredit  The sportCredit. |
| int | coreSelected  The coreCourse has been selected. |
| int | totalGeneralCredit  The total of the GeneralCredit. |
| int[] | generalCredit  The total of the generalCredit. |
| ArrayList<Course> | generalSelected  An ArrayList that holds all general courses has been selected. |
| ArrayList<Course> | sportSelected  An ArrayList that holds all sport courses has been selected. |
| Boolean[] | coreCheck  check if the course is coreCourse. |
| **Constructor** | |
| **NCCUStudent(int stdID, int grade, int department)**  Enable to construct a NCCUstudent object with given student id, grade, department.  Initialize generalSelected, sportSelected, generalCredit, coreCheck.  Set sportCredit as 0,set coreSelected as 0, set totalGeneralCredit as 0. | |
| **Instance Methods** | |
| - | 2 getter for 2 attributes(getStdId(), getGrade()) |
| void | addGeneralCourse(Course c)  1. Check whether the course has been selected.  2. If the course has not been selected, add it to the student's course selection list and mark it as selected.  3. Perform different actions based on the category of the course:  - For basic general courses, if the credit has not reached 6 credits, add the credit and update the total general credits.  - For core general courses, if the credit has not reached 7 credits and meets the conditions for core courses, add the credit and update the total general credits.  - For other general courses, if the credit has not reached 3 credits, add the credit and update the total general credits.  4. If the course has already been selected, do not perform any operations. |
| void | addSportCourse(Course c)  Add the selected sport course to the student's list of sport courses and set the sport credit to 1. |
| void | selectedList(int misSelectedCredit, boolean[] misCheck, MISCourses misCourses)  Print a summary of the selected courses for a student, including general courses, sports courses, and information about the credits taken in each category. Additionally, it displays information about the remaining required courses, including compulsory courses that haven't been selected, courses from miscellaneous categories, and the shortage of credits in specific general education categories. The output provides a comprehensive overview of the student's course selections and the status of their progress toward fulfilling various academic requirements.(See output below) |
| void | selectedList(int misGraduateSelectedCredit, boolean[] misGraduateCheck, GraduateMISCourses graduatemisCourses)  Print a summary of the graduate student's selected courses, including the selected graduate courses, the total credit fee, and a list of unselected graduate courses. The output provides a comprehensive overview of the student's progress in graduate studies, including earned credits, total credit fees, and a list of graduate courses not yet selected. |

3. Create **MISStudent** class

| **MISStudent** | |
| --- | --- |
| **Modifier and type** | **Method (or Variable) and description** |
| **Instance variable** | |
| int | misSelectedCredit  The credit has been selected. |
| boolean[] | misCheck  Check if the student is MISSStudent. |
| ArrayList<Course> | misSelected  An ArrayList that holds all courses has been selected. |
| String | departmentName  The name of the department. |
| **Constructor** | |
| **MISStudent(int stdID, int grade, int department)**  Enable to construct a MISStudent object with given stdID, grade, department from superclass (NCCUStudent).Initialize the misCheck, misSelected.Set departmentName as "資訊管理學系",misSelectedCredit as 0. | |
| **Instance Methods** | |
| void | addMisCourse(Course c)  Add a MIS course to the list of selected courses for the student. |
| void | selectedList(MISCourses misCourses)  Print a summary of selected MIS courses, including course name, credits, schedule, instructor, and course code. It also calls the selectedList method from the superclass (NCCU Student) to print additional information about general courses. |
| void | addMisCredit(Course c)  Add the credits of a MIS course to the total credits of MIS courses for the student. It checks if the course has already been added to avoid duplicate credit calculations. |

4. Create **CourseFactory** class

Design Pattern:

Use multiple factory methods (createGeneralCourese, createRequiredCourse, createSportCourse, createCourse), which accept parameters that allow customization of the created Course object.

This enables flexibility in creating courses with different attributes.

| **CourseFactory** | |
| --- | --- |
| **Modifier and type** | **Method (or Variable) and description** |
| **Instance variable** | |
| - | - |
| **Constructor** | |
| **-** | |
| **Instance Methods** | |
| Course | createGeneralCourse(String name, int credit, int id, String time, String teacher)  Create non-core general courses with given name, id, time, teacher. |
| Course | createGeneralCourse(String name, int credit, int id, String time, String teacher, boolean isCore)  Create core general courses with given name, credit, id, time, teacher, isCore. |
| Course | createRequiredCourse(String name, int credit, int id, String time, String teacher)  Generate required courses with given name, credit, id, time, teacher. |
| Course | createSportCourse(String name, int credit, int id, String time, String teacher)  Generate Sport courses with given name, credit, id, time, teacher. |
| Course | createCourse(String name, int credit, int id, String time, String teacher)  Generate graduate-level courses with given name, credit, id, time, teacher. |

5. Create **MISCourses** class

| **MISCourses** | |
| --- | --- |
| **Modifier and type** | **Method (or Variable) and description** |
| **Instance variable** | |
| ArrayList<Course> | required  An ArrayList that holds all required courses of MISStudent.. |
| CourseFactory | courseFactory  The CouseFactory object. |
| **Constructor** | |
| **MISCourses()**  Initialize the required, courseFactory.  Use CourseFactory to create required courses. | |
| **Instance Methods** | |
| void | addMisCourse(Course c)  Add a course to the ArrayList(misSelected) . |
| void | selectedList(MISCourses misCourses)  Generates and prints a formatted list of selected miscellaneous courses for a student. It displays information such as the student's ID, department, and grade, followed by the selected courses. The details of each selected course include its name, credit, time, teacher, and course ID. If no miscellaneous courses are selected, a message indicating the absence of selected courses is printed. Additionally, the method invokes the selectedList method from the superclass to display further details such as selected credits and specific courses tracked by the system. |
| void | addMisCredit(Course c)  Updates the selected credits for a student based on a given course. It calculates the index of the course in the course array using the course ID. If the course has not been previously checked, the method sets misCheck[index] to true, indicating that the course has been processed. It then increments the misSelectedCredit by the credit value of the given course. |

6. Create **SelectMenu** class

| **SelectMenu** | |
| --- | --- |
| **Modifier and type** | **Method (or Variable) and description** |
| **Instance variable** | |
| MISCourses | misCourses  It allows access to methods and data related to these courses. |
| GeneralCourses | generalCourses  It allows access to methods and data related to general courses. |
| SportCourses | sportCourses  It allows access to methods and data related to sport courses. |
| GraduateMISCourses | graduatemisCourses  It allows access to methods and data related to graduate-level information system courses. |
| **Constructor** |  |
| - |  |
| **Instance Methods** | |
| void | selectMenu(MISStudent misStd, Scanner sc)  this method aim to handle the course selection process for MIS students.It prompts the users to input the desired course category, displays the corresponding course list, and allows the users to add selected courses to the MIS student’s course list. The recursive calls facilitate continuous course selection until the users choose to exit. |
| void | selectMenu(GraduateStudent graduatemisStudent, Scanner sc)  this method aim to manage the course selection process for graduate-level student.It prompts the users to input the desired course category, displays the corresponding course list, and allows the users to add selected courses to the graduate-level student’s course list. The recursive calls facilitate continuous course selection until the users choose to exit. |
| void | selectMenu(OtherStudent otherStd)  this method aim to handle course selection for students other than MIS and graduate-level students.It prompts the users to input the desire course category and display relevant course lists. |

7. Create **SportCourses** class

| SportCourses | |
| --- | --- |
| **Modifier and type** | **Method (or Variable) and description** |
| **Instance variable** | |
| ArrayList<Course> | sport  The ArrayList holds instances of the Course class, representing sports-related course. |
| CourseFactory | courseFactory  used for creating sports courses |
| **Constructor** | |
| **public SportCourses()**  Initializes the sport ArrayList and courseFactory.  Use the courseFactory to create and add sports courses to the sport list. | |
| **Instance Methods** | |
| void | courseList()  Display a list of available sports courses.  Prints a formatted list of sports courses, including courses’ details.It prompts the users to input the course ID of the sports course they want to select. |
| Course | findCourse(int id)  Find and returns a sport course based on the provided course ID. Iterates through the sport list to find a course with a matching ID, if found and not selected, returns the course; otherwise, returns a new Course instance with a flag indicating it’s empty. |

8. Create **GeneralCourses** class

| **GeneralCourses** | |
| --- | --- |
| **Modifier and type** | **Method (or Variable) and description** |
| **Instance variable** | |
| ArrayList<Course> | generalCH  ArrayList to store categories of Chinese courses. |
| ArrayList<Course> | generalEN  ArrayList to store categories of English courses. |
| ArrayList<Course> | generalHumanities  ArrayList to store categories of humanities courses. |
| ArrayList<Course> | generalSocial  ArrayList to store categories of social courses. |
| ArrayList<Course> | generalScience  ArrayList to store categories of science courses. |
| ArrayList<Course> | generalComputer  ArrayList to store categories of computer courses. |
| **Constructor** | |
| **GeneralCourses()**  Initializes ArrayLists for different categories of general courses.  Adding non-core and core general courses in the ArrayList separately. | |
| **Instance Methods** | |
| void | courseList()  Prints a formatted list of general courses for each category, including details such as name, credit, time, teacher, and ID.  Prompts the user to input the course code for the general course they want to track. |
| Course | findCourse(int id)  Searches for a general course in each category based on the provided course ID.  Returns the found course if it exists and has not been selected, if the course has already been selected or the ID is not found, it returns a special Course object. |

9. Create **GraduateMISCourses** class

| **GraduateMISCourses** | |
| --- | --- |
| **Modifier and type** | **Method (or Variable) and description** |
| **Instance variable** | |
| ArrayList<Course> | graduateMISCourses  An ArrayList that stores instances of graduate MIS courses. |
| CourseFactory | courseFactory  Used for creating instances of courses. |
| **Constructor** | |
| **GraduateMISCourses()**  Initializes the graduateMISCourses ArrayList with instances of graduate MIS courses. | |
| **Instance Methods** | |
| - | 1 getter for 1 attributes(getGraduateMISCourses()) |
| void | courseList()  Prints a formatted list of graduate MIS courses, including details such as name, credit, time, teacher, and ID.  Prompts the user to input the course code for the graduate MIS course they want to track. |
| Course | findCourse(int id)  Searches for a graduate MIS course in the graduateMISCourses ArrayList based on the provided course ID.  Returns the found course if it exists and has not been selected, if the course has already been selected or the ID is not found, it returns a special Course object. |

10. Create **GraduateStudent** class

| **GraduateStudent** | |
| --- | --- |
| **Modifier and type** | **Method (or Variable) and description** |
| **Instance variable** | |
| int | CreditFee  Represents the cost per credit for graduate students. |
| int | totalCreditFee  The total credit fee for graduate students. |
| int | misGraduateSelectedCredit  Tracks the total number of credits selected by the student. |
| boolean[] | misGraduateCheck  To check whether a specific course category has been selected. |
| ArrayList<Course> | misGraduateSelected  To store the graduate courses selected by the student. |
| String | departmentName  Represents the department name, set to "資訊管理學系." |
| **Constructor** | |
| **GraduateStudent(int stdID, int grade, int department)**  Calls the constructor of the superclass NCCUStudent to initialize basic student information. Initializes specific attributes related to graduate students. | |
| **Instance Methods** | |
| - | 2 getter for 2 attributes(getCreditFee(),getTotalCreditFee()) |
| void | calculateCreditFee()  Calculates the total credit fee based on the selected graduate MIS credits and the cost per credit. |
| void | addGraduateCourse(Course c)  Adds a graduate MIS course to the list of selected courses. |
| void | selectedList(GraduateMISCourses graduatemisCourses)  Prints a list of selected graduate MIS courses, including details such as name, credit, time, teacher, and ID.  Calls the selectedList method from the superclass NCCUStudent to display additional information. |

11. Create **main** class

In the main class,

a.User Input and Student Initialization:

1.Prompts the user to input a student's ID, grade, and department code.

2.The user is informed about the department codes for different departments (e.g., 306 for MIS, 356 for MIS graduate, 0 for other departments).

3.Based on the user's input, an instance of either MISStudent, GraduateStudent, or OtherStudent is created.

b.Menu Selection:

1.Creates an instance of the SelectMenu class.

2.The selectMenu method of the SelectMenu class is then called with the appropriate student object and a Scanner object to manage the menu selection.

c.Loop and Error Handling:

1.Use a loop to ensure correct user input for student information.

2.If the user enters an invalid input (e.g., a non-integer), an error message is displayed, and the user is prompted to input the correct information.

d.Department Code Validation:

1.Check the department code entered by the user and create an instance of the appropriate student type.

2.If the department code is not recognized, the user is prompted to enter the correct department code.

e.Exception Handling:

1.Use try-catch block to catch any exceptions that might occur during user input or object creation.

| **Sample Output** |
| --- |
|  |

| **Sample Output** |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |