**Object-Oriented Programming Course Design Task Guide**

**（Second Semester, Academic Year 2023/2024）**

**Instructor：**

### Task Instructions

Each student may independently complete.

The comprehensive score for the course design consists of two parts: "Design Report 50% + Program Demonstration 50%". The scoring criteria for each part are as follows:

1. **Design Report Assessment Objectives and Standards**

|  |  |  |
| --- | --- | --- |
| **Course Objectives** | **Scoring Criteria** | |
| Target 1 | Ability to use OOP design concepts to design classes and the relationships between classes. The design of the classes is rational, the business logic is complete, the diagrams are standardized, the format meets the requirements, and there is a detailed analysis and summary. | Excellent  （90-100） |
| The design of the classes is slightly unreasonable with minimal issues, the business logic design is not fully complete, the diagrams are standardized, the format meets the requirements, and there is an analysis and summary. | Good  （75-89） |
| The design of the classes has many unreasonable aspects, the business logic design is incomplete, the diagrams are not standardized, the format basically meets the requirements, and there is no analysis and summary. | Average/Pass（60-74） |
| Unable to use OOP design concepts to design classes and the relationships between classes, the design of the classes is completely unreasonable, the diagrams are not standardized, the format does not meet the requirements, and there is no analysis and summary. | Fail  （0-59） |

1. **Program Demonstration Assessment Objectives and Standards**

|  |  |  |
| --- | --- | --- |
| **Course Objectives** | **Scoring Criteria** | |
| Target 2 | Use a graphical user interface or command-line interface to receive and display data, capable of using a database or files to store data. Able to write a C++ program that meets the requirements, is runnable, and follows OOP design principles. The code is well-organized, the business logic is reasonable, and the interface interaction is good. | Excellent  （90-100） |
| Uses a command-line interface to receive and display data, capable of using a database or files to store data. Able to write a C++ program that meets the requirements, is runnable, and follows OOP design principles. The code is quite standardized, the business logic is reasonable, and at least 90% of the functional requirements are implemented, with good interface interaction. | Good  （80-89） |
| Uses a command-line interface to receive and display data, capable of using a database or files to store data. Able to write a C++ program that meets the requirements, is runnable, and follows OOP design principles. The code is quite standardized, and at least 80% of the functional requirements are implemented, with good interface interaction. | Average（70-79） |
| Uses a command-line interface to receive and display data. Able to write a C++ program that meets the requirements, is runnable, and follows OOP design principles. The code is quite standardized, and at least 70% of the functional requirements are implemented. | Pass（60-69） |
| Uses a command-line interface to receive and display data. Able to write a C++ program that meets the requirements, is runnable, and follows OOP design principles. The code is not well-organized, and at least 60% of the functional requirements are implemented. | Fail  （0-59） |

The completion degree of the functions is mainly evaluated based on the completion of the step-by-step tasks provided in the task book. It is important to note that, in addition to the first task point, each task point includes the tasks of all previous task points. For example, if the third task point is for a score of 80 or above, then it is necessary to complete the tasks of the first, second, and third task points simultaneously to achieve a functional completion score of 80 or above.**Task**

**[Title] Restaurant Reservation System**

**[Purpose]**

Based on the life experience of making reservations at a restaurant and waiting on-site, design a restaurant reservation system with a clear and easy-to-operate interface. Train the ability to comprehensively apply the knowledge learned to solve practical problems. It is required to use object-oriented (OOP) thinking and its main characteristics in the analysis and design, and to implement the designed system through object-oriented programming. By actually using object-oriented mechanisms, strengthen the concept of object-oriented analysis, design, and programming, improve the ability to analyze and solve problems, and enhance the level of program design and debugging.

**[Requirements]**

* Each student must complete the task independently.
* The course design period is 1 week.
* Students must ask for leave if they need to leave the school for some reason. Absence without a valid reason during the course design period will be treated as truancy; those absent for more than a quarter of the time or those who do not submit the experimental report as required will fail the course.
* Analyze, design, and implement the system using the object-oriented programming design method. The main body of the implementation code must be in C++. Reasonably abstract and simplify the main functions, roles, resources, etc., involved in the system during the analysis phase, encapsulating the relevant attributes and operations into corresponding classes; use inheritance and composition reasonably to reduce redundancy, making the system highly cohesive and loosely coupled; on the basis of inheritance, reasonably use polymorphism, operator overloading, and even templates and exception handling, making the system easy to extend and maintain.
* The interface should be applicable, the code files should be organized reasonably, the naming should be standardized, the comments should be concise and appropriate, and the course design report should be clear and logical.

**[Content Description]**

Restaurants are an ideal place for people to have meals together, business meetings, or celebrations. With the acceleration of modern life pace, time has become particularly precious, so more and more people choose to make reservations in advance to ensure a smooth dining experience. In order to ensure customer satisfaction and smooth dining, the restaurant has formulated a series of seat reservation regulations. First, the restaurant will set a time limit for seat reservations to ensure that the meal can prepare ingredients; secondly, to avoid seat waste and improve the accuracy of reservations, customers need to pay a certain amount of deposit, which will be deducted when actually dining, or will not be refunded if the reservation is not canceled in advance; customers must cancel the reservation a certain period of time in advance. This course design is to develop a simple reservation system for a restaurant. The system users are divided into two roles: customers and system administrators. The system administrator maintains and manages information such as private room tables, meals, and reservation situations. Customers can make reservations in advance through the system's reservation function and enjoy a pleasant dining experience.

**[Functional Requirements]**

The system administrator logs in to the system with the assigned account and password, completing the configuration of reservation information, query, and statistical functions; if the customer is logging in to the system for the first time, they need to register, and after completing the registration, they can log in to the system based on the phone number and password, and complete functions such as reservation, viewing reservation results, canceling reservations, and modifying reservation information through the system.

1. Pass (≥60 points) basic requirements: Complete the advance reservation module. Customers log in to the system with preset usernames and passwords, and after logging in, they need to bind visitor information (name, phone number, etc.). Customers can make reservations for the next day's dining seats within the specified time period of the day, and the reservation includes: reservation dining time (the restaurant can specify the time), reservation of table type (such as private room, table for four, table for two, etc.), if there is no available reserved type of dining seat, the reservation will not be successful; customers need to pay a certain amount of deposit (simulated operation); customers can obtain a reservation order after a successful reservation, and the reservation order includes reservation order number, reservation dining time, private room number or table number, etc.
2. Medium (≥70 points) basic requirements: On the basis of 1, increase (1) the ability to cancel reservations and modify reservation information (three hours in advance) (2) implement the registration function for visitor users, the registration information includes name, phone number, and password, etc., and a simple verification of the validity of the phone number is required when registering, such as length restrictions on the phone number. (3) Customer information is saved in file form, and after the system restarts, it can implement login verification and information retrieval based on the information in the file.
3. Good (≥80 points) Basic Requirements: On the basis of 1 and 2, the following enhancements are made:
   1. When making a reservation for a dining seat, a reservation for meals (set menu or à la carte) is also made simultaneously. A deposit of 30% of the total price of the meal is required, and the reservation order includes an additional list of meals. Successful reservation comes with a total price discount.
   2. The system administrator has the capability to delete, query, and modify dining seats, meals, and reservation orders. All data that has been added, deleted, or modified can be saved in file format, and the system can complete the reservation function based on the modified data after a restart.
4. Excellent (≥90 points) basic requirements: On the basis of 1, 2, and 3, the system functions are complete, the operation is clear and convenient, and one of the following conditions is met: (1) A database is used (2) A graphical user interface is used (3) There are new functions or highlights, such as completing the on-site queuing module. Customers can obtain a dining queue number for the day within the specified time of the restaurant on the same day (10-12:30 and 17:00-19:30), and the queue number includes the queue number, table type, waiting quantity, etc.; the system administrator can delete and query the queue number and set the quantity of the queue number.

**[Reference Materials]**

For graphical user interface (GUI) development option 1 - Qt

Qt is a cross-platform C++ graphical user interface application development framework that can develop both GUI programs and non-GUI programs, such as console tools and servers. Qt is an object-oriented framework that is easy to expand and provides a rich API, including more than 250 C++ classes.

It is best to install the qt-vsaddin-msvc2019-2.4.0vsix plugin with vs. Reference materials are as follows:

Official website: https://www.qt.io/zh-cn/develop

Tutorial: http://c.biancheng.net/qt/

Teaching video: https://www.bilibili.com/video/BV1Jp4y167R9/?spm\_id\_from=333.337.search-card.all.click

Graphical user interface (GUI) development option 2 - SMFL

SFML (Simple and Fast Multimedia Library) is a multimedia library that provides a simple interface for each component of the PC to simplify the development of games and multimedia applications. It mainly consists of five modules: system, window, graphics, audio, and network. For details, please refer to the website:

https://github.com/ocornut/imgui

https://github.com/SFML/imgui-sfml

It is recommended to use a file database - sqlite

https://blog.csdn.net/QIJINGBO123/article/details/86663902

CPPsqliteDemo\_VS2019.7z

sqlitewin32\_staticlib\_VS2019.7z

Data persistence.pdf

Use of singleton pattern.pdf