

A Project-based Curriculum for Teaching C++ Object-Oriented Programming

Yen-Lin Chen¹

Chuan-Yen Chiang²

Yo-Ping Huang³

Shyan-Ming Yuan²

¹Department of Computer Science and Information Engineering,
National Taipei University of Technology, Taipei, Taiwan
Email: ylchen@csie.ntut.edu.tw

²Department of Computer Science
National Chiao Tung University, Hsinchu, Taiwan

³Department of Electrical Engineering,
National Taipei University of Technology, Taipei, Taiwan

Abstract—This paper presented a project-based remedial curriculum for teaching the C++ programming language and object-oriented programming (OOP) skills and concepts. The pedagogical approach of the proposed curriculum comprises of a set of homemade projects for facilitating the students to quickly learn the essential C++ and OOP skills by accordingly implementing a large-scale communication software system. Based on the constructivist learning technology, the proposed project-based curriculum can effectively improve and enhance the learning effectiveness and interests of the students via the hands-on, minds-on, and learning by doing practices related to their lives. From the students' survey and grade assessment results, the proposed project-based curriculum and practical homemade projects demonstrates its effectiveness and feasibility for motivating the students' interests on C++ and OOP skills and incorporating these skills for developing practical and large-scale software.

Keywords— Computer science education; project-based curriculum; communication software; object-oriented programming (OOP)

I. INTRODUCTION

In the contemporary information and communication technology (ICT) industries, programming languages and object-oriented programming (OOP) has become important and essential skills for the engineers graduated from computer science (CS) related departments. In this situation, programming courses are mostly issued in the undergraduate curricula of computer science related departments [1][2]. While the fundamental OOP features are mostly taught in conventional OOP courses, the abilities on building up a large-scale software system to solve the practical and technical problems are still difficult for students to learn and intensify. Therefore, a practical lab-based or project-based curriculum is necessary for developing students' abilities on accomplish some large-scale software by incorporating various OOP programming skills [3].

Due to the industrial trends of the information and communication technology (ICT) in the authors' country, the industrial employers mostly need the software engineers having the programming abilities on communication

software, embedded system software, device drivers, and multimedia signal processing applications of the consumer electronics products. In this situation, the C++ programming language are essential for the students to seek for technical jobs, because of the C++ language technicalities can not only provide high level programming features such as object-orientation paradigms and generic programming, but also can offer low level programming features such as bit-wise and device I/O control abilities [5][6]. Although the steep learning curve of C++ programming language may obstruct the students' interests and motivations during their course times, the existent studies on the four-year undergraduate computer science curriculums, such as CC2001 [1][2] and project-based programming laboratory courses [4], can offer sufficient trainings for students to practice their C++ programming and OOP skills.

However, in each academic year, many students (including domestic and international students) who held non-CS bachelor degrees are admitted into the graduate program in the authors' department, i.e. the department of computer science and information engineering of the National Taipei University at Taiwan. These non-CS admitted students may took some basic programming courses from other engineering-related departments, but mostly lack of sufficient CS background and cannot have sufficient trainings on C++ and OOP programming skills for their graduate studies and thesis works. Therefore, to catch up with the requirement of the programming skills for these students without CS background, a supporting and remedial curriculum is needed for quickly strengthening their C++ programming and efficiently training their software design skills.

To accomplish the aforementioned demands, this study design a project-based curriculum to rapidly instruct these students with major C++ and object-oriented programming (OOP) skills, as well as assign them to implement a set of consequent homemade projects, which will eventually construct a large-scale software system. For assisting their graduate thesis studies, the objective of the propose curriculum mainly focus on let the students to learn essential OOP concepts and problem-solving skills, rather than just

teaching a programming language. During their thesis studies, the students may need some convenient tools to collect the communicated information, share their discussion records and research materials with their class and lab mates, or advisors. Therefore, through a set of consequent homemade projects with various programming solutions to a consistent objective, that is, building-up their own multimedia communication application for supporting their thesis studies, students can rapidly develop their problem-solving skills and learn how to apply practical solutions using different OOP concepts. As can be seen from the students' survey and grade assessment results, the proposed project-based curriculum and practical homemade projects demonstrates its effectiveness and feasibility for motivating the students' interests on C++ and OOP skills and incorporating these skills for developing practical and large-scale software.

II. CURRICULUM DESIGN AND HOMEMADE PROJECT ASSIGNMENTS

In this section, we introduce our project-based curriculum design for students to effectively and faithfully exercise and practice all the general and important OOP issues on designing and accomplishing multimedia communication software. This goal can be achieved by the class lectures for major elements of OOP, as well as a correlated homemade project assignments. The overall task for accomplishing a multimedia communication software is arranged as seven successive project assignments, and so that the students can gradually design and implement a large scale software system accomplished with their learned OOP programming skills. As a result, the students will achieve a full-functioned multimedia communication software at the end of the curriculum. This proposed curriculum is arranged into seven major lecture topics and seven homemade project assignments, as tabulated in TABLE I. .

In the semester, we spend the first one month to introduce the basic C++ programming skills and input and output programming issues, including general topics of basic data types, variables, string processing, numerical computations, error handling, functions, managing source code files, I/O streams, and file I/O processing skills [5][6]. Then, we spend a majority of the class lectures to instruct the major topics of object-oriented programming issues, accompanying with the topics of OO-based GUI programming libraries and software engineering issues [7][8], are introduced to the students. This includes issues on class design, encapsulation, inheritance hierarchy, polymorphism, and using opensource GUI tool libraries (Qt and FLTK libraries) to assist the students learn large frameworks with OO paradigm, as well as teaching the students to apply software engineering tools and concepts to build up their own software systems. Next, to help the students to manage large amount of data and reusability of code components in large software systems, we also instruct the data structures and generic programming skills with the standard template library (STL) [9]. Finally, the students are asked to apply their learned OOP skills and software engineering concepts

to accomplish their own multimedia communication software systems as the achievements of taking this course.

The homemade projects assigned in the proposed curriculum are addressed as follow.

A. *Project Assignment 1: Implementing Text-Parsing Application*

In this first homemade project, students are asked to implement text-parsing application for practicing the fundamental C++ programming skills. This text-parsing application need have the functions to record and parse the text contents of a real-time interactive chat, including the names of the members having joined the chat, their corresponding talk message sentences typed in the chat console, and the time records of each talk message given by someone joined in this chat. In order to facilitate chat recording, students are also requested to accordingly transform and save the chat records into the XML formatted file. The XML file should also be able to be correctly loaded and parsed for displaying the formatted chat records content on the console mode user interface.

To implement this simple text-parsing application, students can practice the required basic programming skills, including string processing and manipulation, function design, basic error handling, file and console mode I/O stream processing. As a result, students will build up a basic software application with file I/O and text string parsing functions, and a console-mode user interface.

B. *Project Assignment 2: Text Searching & Retrieval Functions*

In the second homemade project, students have chances to implement new fundamental functions of the messenger communication system based on the previous project they had done, including a searching function with simple query syntax. The goal of this project is to provide the information searching and retrieval functions on the desired textual data from the XML files of the archived chat records made in the previous project. A searching syntax is needed to be implemented to perform information retrieval functions on different fields of message records (including message speaker and receiver, message contents, and message timing records) by using field tags and logical operators along with search keywords.

Accordingly, the students are requested to implement the above-mentioned search and retrieval functions by parsing and processing the XML files of chat records, and integrate these functions with an enhanced console user interface to response the user command lines and display the processing results.

C. *Project Assignment 3: Using OOP Skills and Exception Handling*

In the third homemade project, students will have the chances to practice the OOP skills and concepts to rework and re-organize their built-up programs in the two previous projects, and thus the students can learn how to solve the programming problems using essential OOP skills.

Accompanying with the course lectures on the basics of object-orientation and Universal Modeling Language (UML) class diagrams, students are asked to convert their procedural codes, i.e. the text parsing and searching functions implemented in previous two assignments, to OO implementation. Moreover, students are also required to implement some exception handling classes in their built-up classes to provide error message reporting and handling functions. As the code fragments shown in Figure. 1(a), students should implement a class “chat” that integrate the text file processing and XML parsing functions they implemented in the 1st assignment. Then students are required to design a set of hierarchical classes to provide different ways of searching functions, such as searching the chat records by time, name, and message contents, by implementing a base class “DoSearch” as well the subclasses “DoTimeSearch”, “DoNameSearch”, and “DoMessageSearch” for conducting specific searching functions by practicing the OO skills of inheritance and polymorphism, as depicted in Figure. 1(b).

D. Project Assignment 4: Building up a Graphical User Interface (GUI)

Students are often interested in developing GUI applications, which can be conveniently implemented via the Qt opensource GUI tool library with OO programming concepts. In the course lectures, several sample programs are provided for students to quickly familiarize how to apply the essential Qt classes and APIs for building their own GUI applications. Moreover, the Model-View-Controller (MVC) design pattern is also introduced for students to elaborate their GUI applications. Through this practice, students can have a deeper understanding of the OOP concepts on designing a software system that will interact with various users.

In this homemade project, students are required to implement a set of cooperative windows with corresponding GUI widgets, including chat windows, chat invitation windows that will interact with each other, and content search windows. First, the students need to design their main chat windows with text input and output, a “submit” button, and tool menu with required action items, where the submit button and functional items should be appropriately enabled and disabled under different conditions.

E. Project Assignment 5: Implementing Network Communication via Socket Programming

In this project assignment, students can have the chances to practice how to exploit the socket programming techniques to transfer messages and data between the users via the network communication. To implement network communication functions, students may consider integrate the socket classes provided by Qt or MFC class libraries into their projects. Moreover, this project also accompany with the course lectures on the UML diagrams and the Standard Template Library (STL). Thus, students have chances to adopt the use-case and sequence diagrams to figure out the requirements and flows of the network communication functions, as well as apply the appropriate STL containers

and algorithms on solving the message queuing problems associated with the network communication issues.

In this sense, students should implement some protocols and processes for making a connection between users. When a connection for transmitting messages between users is started, the message speaker should send a formatted request string that includes the receiver’s IP addresses and username by the created socket, and then, after receiving the request string, the message receiver should send back a formatted response string that includes the speaker’s IP addresses and username. Thus, a connection between two users is created, and users can start their talking by utilizing formatted strings to specify the speaker, receivers, and message contents and transmitting via the connected socket.

F. Project Assignment 6: Integrating Software Testing and Improving Information Management

Because of the importance of the unit testing in software testing and maintenance issues, this project will ask the students to design and practice the use of the unit testing processes on all crucial functions in their designed classes in the previous projects, as well as re-work their codes to let their designed software be testable. Therefore, all the functions of content searching and text parsing functions, network communication, message archiving, and any other facilities implemented in their previous projects are needed to be automatically tested and verified via a systematic unit testing program, which can be implemented by the Qt Unit Test modules provided by the Qt Library. When the user executes the testing program, the program should accordingly start to execute each test function, and no anymore commands are needed for the testing process.

Moreover, accompanying with the issues on improving the efficiency and reliability of their message communication software, students are also asked to apply some advanced data structures and language features. In this project, they are also practice to adopt the associative containers, i.e. the map containers, to re-write the corresponding classes and algorithmic functions of message record searching and retrieval in their program, and so that these functions can provide more efficient message archiving performance. After completing the new program with the improved message searching and retrieval functions, students can again practice the software testing process, and they can have the chances to learn how to effectively verify the new version of the program when some classes and functions are updated. Having the complete testing program experiences, the students can efficiently improve and maintain their implemented message communication software with a more skillful and elegant way.

G. Project Assignment 7: Completing a Messenger Communication Software

The purpose of the final project assignment is to provide the students a practice chance of integrating all the C++ programming skills and technicalities to fulfill a complete communication software application. At this point, the students should have sufficient understandings on applying and integrating the C++ programming skills and OOP

technicalities, including file processing, string processing, OOP paradigms, GUI programming, generic data retrieval, software testing, and network communication design, to achieve their own large scale real-time communication application. Thus, the students are encouraged to extend the basic messenger communication application by incorporating more additional skills and features, such as multimedia data file transferring, undo/redo functions, and other customized creative designs.

In this final project, students are also encouraged to extend their messenger software projects by their innovative ideas, including more elegant human-computer interfaces (such as automatic IP recording, multi-language supporting, customized font color and size setting interfaces), multimedia chatting interfaces and message record management functions. The implemented classes and functions of the students' resulting messenger communication software are also required to undergo the unit testing processes. The snapshots and the class diagram of the students' messenger communication software are shown in Figure. 2 and Figure. 3. As a result, these extensive practices can help the students have creative skills on apply and integrate multiple opensource projects and their codes to achieve large-scale software projects during this course.

III. RESULTS AND DISCUSSIONS

The remedial C++ object-oriented programming course for the native and international students who held non-CS bachelors began in the Fall 2009 semester at the graduate school of the computer science and information engineering of the National Taipei University at Taiwan, when this course was instructed as the conventional programming course. Started from the Fall 2010 semester, the project-based curriculum for implementing a large-scale practical communication software by incorporating C++ programming and OOP skills is introduced in this remedial course. Various forms of the students' feedback, such as the feedbacks and discussions via the course Facebook social-network forum and project reports, were accordingly adopted to adjust and improve the course contents. In each project assignment, students are encouraged to discuss and report their encountered difficulties and required programming skills and knowledge via the course forum and project reports.

A course survey was performed at the end of the Fall 2011 semester, and all 27 students (including domestic and international students) enrolled in the course at Fall 2011 completed the survey questionnaires. The survey results from five questions are listed in TABLE II. . As depicted in TABLE II. , most students mostly agree strongly that there are significant contributions for them to understand the C++

and OOP concepts and software design skills from the course contents and homemade projects that build up a communication software (TABLE II. , 4.46 out of 5 in question 1, and 4.55 out of 5 in question 2, respectively).

The students also definitely agree that they are interested in the course contents and homemade projects, as well as are encouraged to study hard to enhance their problem solving abilities on developing large-scale software projects (TABLE II. , 4.64 out of 5 in question 3, and 4.36 out of 5 in question 4, respectively). They pay more than 20 hours weekly in average on their homemade projects throughout the whole semester. The remaining question also reflect that the students feel the course social-network forum can efficiently provide intermediate supports and helps from the TAs and the instructor, provide them a feedback way to the instructor to improve the course contents, and share the programming knowledge and experiences (TABLE II. , 4.55 out of 5 in question 5). The survey results demonstrate that the proposed project-based curriculum on accomplish a real-time communication software successfully provoke the students' interests on C++ and OOP skills, and contribute to students learning the advantages of OOP and practical software engineering issues.

ACKNOWLEDGMENT

This work is supported by the National Science Council of the Republic of China under Contract No. NSC 99-2221-E-027-100 and NSC-100-2221-E-027-033.

REFERENCES

- [1] ACM AIS IEEE-CS, "Curricula recommendations," May 15, 2009 [Online]. Available: <http://www.acm.org/education/curricula-recommendations>.
- [2] The Joint Task Force on Computing Curricula, Computing curricula 2001, J. Educ. Res. Comput. 1 (2001), 1–240.
- [3] J. Lang, G. C. Nugent, A. Samal, and L.-K. Soh, Implementing CS1 with embedded instructional research design in laboratories, IEEE Trans. Educ. 49 (2006), 157–165.
- [4] W.-K. Chen and Y. C. Cheng, Teaching Object-Oriented Programming Lab with Computer Game Programming, IEEE Trans. Educ. 50 (2007), 197-203.
- [5] S. Lippman and J. Lajoie, C++ Primer, 3rd ed. Reading, MA: Addison-Wesley, 1998.
- [6] Bjarne Stroustrup, "Programming: Principles and Practice Using C++", Addison Wesley, 2009.
- [7] Unified Modeling Language (UML) Specification, Object Management Group, [Online]. Available: <http://www.omg.org/>
- [8] J. Blanchette, M. Summerfield, "C++ GUI Programming with Qt4, 2/e", Prentice Hall, 2008.
- [9] N. M. Josuttis, "The C++ Standard Library : A Tutorial and Reference", Addison Wesley, 1999.

```

class chat{
    //...

private:
    string time;
    string speaker;
    string reciever;
    string message;

    // ...
};

```

(a) Sample code fragments of the class “chat”

```

class DoSearch{
    //...
public:
    virtual void search() = 0;

protected:
    logic_not();
    logic_and();
    logic_or();
    //...
};

class DoTimeSearch : public DoSearch{
    //...
public:
    void search();
    //...
};

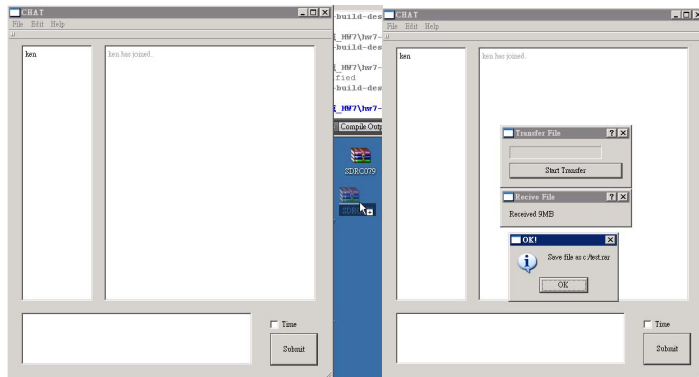
class DoNameSearch : public DoSearch{
    //...
public:
    void search();
    //...
};

class DoMessageSearch : public DoSearch{
    //...
public:
    void search();
    //...
};

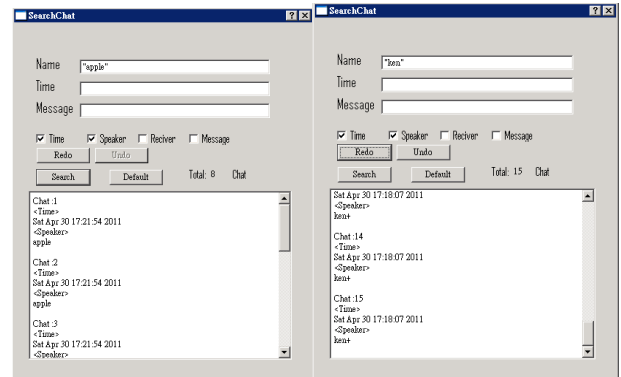
```

(b) Sample code fragments of the base class “DoSearch” and its sub-classes

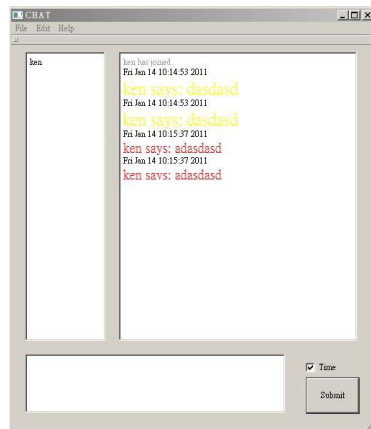
Figure 1. Snapshots of OO implementation of the text parsing and searching functions in the previous homemade projects.



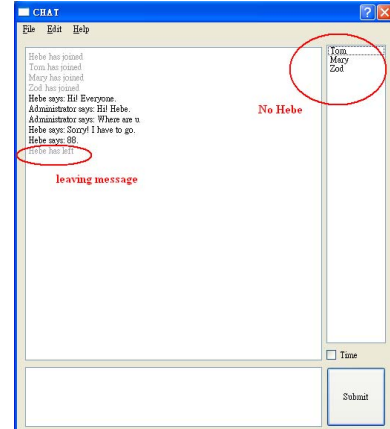
(a). Snapshots of network file transferring applications



(b). Snapshots of undo and redo interfaces of chat record retrieval applications



(c). Snapshots of customized font color and size setting interfaces



(d). Snapshots of improved

Figure 2. Snapshots of the extended messenger communication software implemented in the final project.

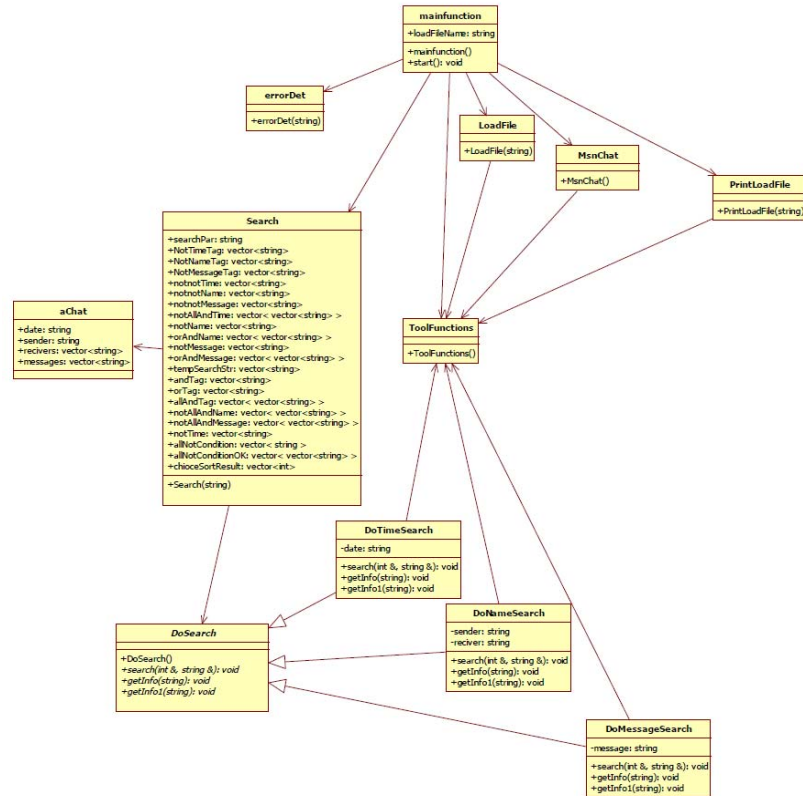


Figure 3. Sample UML class diagram of the students' resultant messenger communication software

TABLE I. CURRICULUM TOPICS AND PROJECT ASSIGNMENTS.

Major Topics	Contents	Homemade Project Assignments
Basic Programming Skills	Basic data types, variables, strings, computations, error handling, functions, source code files	Homemade Project 1
Input and Output Programming	Console mode I/O, file I/O, I/O streams, string processing,	Homemade Project 2
Object-Oriented Programming	Class design, UML class diagrams, encapsulation, inheritance hierarchy, subtype polymorphism	Homemade Project 3
Graphical User Interface (GUI) Programming	GUI tool libraries, GUI widgets, GUI design, event-driven based programming	Homemade Project 4
Generic data processing and network programming	Templates, data containers, algorithms, the Standard Template Library (STL), network communication programming	Homemade Project 5
Software Engineering Issues	Software testing and maintenance, UML modeling diagrams for OOP, Basic Design Patterns	Homemade Project 6
Network and Mobile Communication Software Design	Real-time communication software, advanced GUI implementation issues, software integration	Homemade Project 7

TABLE II. COURSE SURVEY QUESTIONS AND RESULTS FROM STUDENTS ENROLLED IN FALL 2011.

Survey questions (1=Strongly Disagree to 5=Strongly Agree)	Results
1. I fell that I have a better understanding of the C++ and OOP concepts and software design after taking this course.	4.46 / 5
2. The homemade projects can effectively help me to learning the C++ and OOP software design	4.55 / 5
3. I am seriously study hard to develop my programming skills to solve the problems and accomplish large-scale software.	4.64 / 5
4. The course contents interest me and I would like recommend this course to my junior classmates.	4.36 / 5
5. The course social-network forum can help me learning programming skills and conveniently obtain supports from TAs and the instructor	4.55 / 5