“Computer Programming IV” as Capstone Design and Laboratory Attachment

Shoichi Yokoyama†
Yamagata University, Yonezawa, Japan

ABSTRACT
A new obligatory subject, Computer Programming IV, is organized in the Department of Informatics, Faculty of Engineering, Yamagata University. The purposes of the subject are as follows: (1) Attachment to each laboratory for bachelor thesis was usually at the initial stage of the student’s fourth academic year. This subject actually moves up the attachment because students are tentatively attached to a laboratory for this subject. The interval to complete their bachelor thesis is extended by half a year. (2) In each laboratory, students cooperate with each other to complete their project. The project becomes capstone design which JABEE (Japan Accreditation Board for Engineering Education) is recently emphasizing. We not only explain the introduction of this subject, but also report some case studies.

Keywords: Engineering education, Capstone design, Laboratory attachment, Project, JABEE

I. Introduction

The education program of the Department of Informatics, Faculty of Engineering, Yamagata University (YUDI) was accredited in 2003 by Japan Accreditation Board for Engineering Education (JABEE) [1], the second to be accredited for information engineering. Through the intermediate examination in 2005, the program was re-accredited in 2008 for the 2009 to 2014 period.

At the YUDI, we have an Education Quality Improvement Committee (EQIC) organized among the entire staff. At the re-accreditation in 2008, the EQIC grasped that the most important issue for improving education was engineering design. Engineering design is also considered an important issue by JABEE, but before 2008, most of efforts were biased toward the graduation research students engaged in after their attachment to a specific laboratory.

In order to address this issue, the EQIC improved the curriculum and established the new obligatory subject Computer Programming IV. The new curriculum was first adapted to the students entering in 2007. The subject in the curriculum is taught in the second half of students’ third academic year, so that students first took the subject in 2009. The detailed plan created before this first use is described in [2].

The present paper describes the syllabus and proceeds to describe case studies of some laboratories. We explain three years of results.

Computer Programming IV has the following two purposes:

1. Attachment to each laboratory for bachelor thesis was usually at the initial stage of the fourth year. This system is common to all of the Faculty of Engineering and cannot be changed by one department. However, for this subject, students are tentatively attached to the laboratory in which they want to do research and write their thesis. The interval to complete their bachelor thesis is extended for half a year.

2. In each laboratory, students are grouped as described below and cooperate with each other to complete their project. This is part of the capstone design, which JABEE is recently emphasizing. The time for consuming the capstone design was previously biased to the bachelor thesis.

There were previously already three obligatory subjects, Computer Programming I, II, and III. In these subjects,
one tutor teaches a practical programming language such as C or Java with the help of TA staff.

In Computer Programming IV, 5 to 10 students are attached to each laboratory, in which groups of 2 to 4 students complete a project. We will describe the content of this subject in the following sections.

II. Syllabus of Computer Programming IV and Practical Attachment to Each Laboratory

1. Content of Syllabus

In the syllabus of the Faculty of Engineering [3] (written in Japanese), the theme of the subject Computer Programming IV is written as follows: “The main purpose of this subject is to learn application techniques necessary to realize the society for which informatics is the target. The other purpose is to learn solution methods for practical problems through programming as a way of connecting the third year with the fourth, in which the bachelor thesis is written.”

Further, the syllabus describes the goal of the subject as follows: “You should improve on the practical skills which you have already learned through five semesters and gain the basic knowledge necessary for the bachelor thesis. You are attached to a specific laboratory and will learn the background for your thesis through the programming done for this subject. You should take an active role in this subject because it provides you a chance to learn other programming languages than those you have already learned in Computer Programming I, II, and III.”

The goals of the subject are described as follows:

1. Students learn to solve the required application problems using the appropriate programming language(s).
2. Students learn to work independently and systematically in order to complete a project within a set time limit.
3. Students learn logical thinking and presentation through programming.
4. Students learn to take advice from a tutor and develop their own project.
5. Students learn to plan how to solve a problem and carry out effective programming design, coding, and testing.

As keywords, various programming languages (C, LISP, Java, Perl, Python, PHP, PostgreSQL, HPF, Mathematica, and C++) and the motivation to the bachelor thesis are written in the syllabus.

The standard system of evaluation is to measure both attitude to the subject (50 points) and programming ability (design, coding, and testing) (50 points), and 60 points is the minimum passing score.

2. Procedure and Practical Attachment of Students

The procedure of this subject is as follows. First, students explore at the content of each laboratory during the open campus held in August, which is during the beginning of summer vacation (in 2011, open campus took place on August 5th). In the obligatory subject Informatics Training taken during the first half of the second year, students are tentatively assigned to a laboratory at random and visit five laboratories total, including the laboratory they belong to. During the open campus, students can freely visit any laboratory and consider their future.

Second, each student expresses his/her wishes to a laboratory for attachment. Depending on his/her grade point average score, (s)he is attached to the laboratory and usually writes his/her bachelor thesis in that laboratory.

Third, at the start of the sixth semester (in October), student guidance takes place in each laboratory and students decide their project.

III. Case Studies

In this Section, we present the case studies in each laboratory. Table 1 lists the case studies. Laboratories

<table>
<thead>
<tr>
<th>Table 1 Structure of Section III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec.</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>
take their name and basic field from their tutor. Each laboratory’s url can be accessed through the YUDI web page (http://www.eie.yz.yamagata-u.ac.jp/i/jp/).

1. Case Study 1: Yokoyama-Inoue Lab.

In this laboratory, Professor Shoichi Yokoyama researches natural language processing and Assistant Professor Masashi Inoue researches interaction engineering. The purpose here is for students to learn practical programming planning, design, and testing relative to the field.

A. Results and Instructor Impressions for 2009

In 2009, the first year of this subject, nine students came to our laboratory and were divided into three groups of 3 students. We introduced new computers and asked each group to install the necessary software, for example, well-known freeware programs for morphological analysis such as Chasen (developed by NAIST), JUMAN (Kyoto Univ.), and MeCab (Kyoto Univ. and NTT).

After software installation, each group wrote programs to obtain statistics and occurrence frequencies from texts. Statistics are obtained, for example, for the number of letters in each sentence, the number of morphemes and/or words in a sentence, co-occurrence relations of morphemes, and occurrence frequencies of words. Two groups adopted the programming language Ruby, and one group adopted Perl. The results depended on the programming language: the group that adopted Perl had relatively better results than the groups that adopted Ruby because of the time required to learn the Ruby language.

Each group made a presentation at the end of the semester.

B. Results and Instructor Impressions for 2010

In 2010, based on the results in 2009, the practical was carried out in the common computer room because it was more convenient for developing and providing access to subject materials. Ten students came to the laboratory and were divided into three groups of 3 or 4 students. We first introduced Perl programming for simple text processing and then wanted each group to learn both a morphological freeware language and R, which is a language used for statistics. We observed that using the R programming language made graphical representation easier.

Each group made a presentation at the end of the semester.

C. Results and Instructor Impressions for 2011

In 2011, the practical was also carried out in the common computer room. Nine students came to our laboratory and were divided into three groups of 3 students. First, as an introduction and program exercise, each student made his/her own program to calculate the number of credits and determine pass or fail for going on to the stage of working on the bachelor thesis. Second, each group developed a program providing tools for natural language processing, including determining the number of words in a sentence, co-occurrence relations of words, and occurrence frequency of words. This year, R and Python were used as programming languages and each group gave a first presentation.

After the first presentations, members were shuffled between groups and each new group selected a theme from among a choice of several related to natural language processing. This semester, one group selected semantic implication of sentences, another group selected the derivation of important sentences, and the third group selected the arrangement of sentences. Each group completed their project and performed a second presentation.

2. Case Study 2: Goto–Tada Lab.

In this laboratory, students first learned Visual C++ and then pairs of students developed various programs. Each pair decided for itself the theme and design of the program. Over two years, students developed a variety of programs, including an inventory control system for books, a credit control system for the university, an extended version of petty cashbook, and a property management system. The basic data processing of these programs was good, but most of the programs varied little from those in the textbook.

In this laboratory, students were divided into three groups, each of which studied a different theme. The C programming language was used. One group studied "speaker indexing", in which the speaker label was attached for the speech data uttered by people, and developed a program for calculating the distance of speaker models. Another group studied "speaker adaptation", in which the characteristics of specific speakers are imported and the performance is incremented. This group developed a tool for investigating the parameter variation between before and after adaptation. The third group studied an automatic evaluation system of English pronunciation uttered by Japanese. This group developed a program which gathered evaluation results automatically and calculated the correlation between subjective and objective estimation. Part of their results were presented at an international conference [4].

4. Case Study 4: Koyama Lab.

In this laboratory, students were divided into three groups: the network protocol group, network communication group, and network service group. The network protocol group carried out the design and development of a network simulation program by using a network simulator by referring to a sample program. The network communication group carried out the design and development of a network communication program using the Java language, also by referring to a sample program. The network service group carried out the design and development of a remote communication system by using FLASH action script, again, by referring to a sample program.

5. Case Study 5: Kamitani Lab.

In this laboratory, student’s missions were as follows: to be able to perform structured programming, to successfully develop a program with hundreds of steps by correctly designing its structure, to resolve runtime errors as well as compile-time errors by themselves, and learn how to use the various functions of subroutine libraries. Most practically, the students learned computer graphics programming by using the PGPLOT Subroutine Library and programming languages such as HPF, Fortran90, and Mathematica.


In this laboratory, students were divided into groups of 3 students and each group selected one of three missions: the observation of information propagation in the social media, the investigation of the relation between the length of a program and its runtime, and the improvement of the situation display system for the contents of a web server. In each group, each member took a specific role, such as project manager, programmer, or presenter.

7. Case Study 7: Nomoto Lab.

In this laboratory, students studied the theme “the solution method for the travelling salesperson problem using a genetic algorithm”. The tutor first lectured some basic methods, and then each group designed and developed their program. For the genetic algorithm, the given parameters governed the results and the method itself was very difficult to use. Each group needed to study many trials to achieve improvement.

8. Summary

The grouping method and programming design, development, and testing are common to all the laboratories. Some laboratories assign integrated missions and other laboratories give each group a separate mission. In every group, discussion is necessary: graduate students advise the groups, and in many cases provide lectures.

IV. Conclusion

The structure and results of the obligatory subject Computer Programming IV are described. Three years of implementation is not much from which to make an evaluation, but the programming skills of students were improved relatively more than before the new subject
was added, and the matching of each student to each laboratory improved. In the future, we will investigate the results of additional years and make changes to improve the subject contents.

I acknowledge my colleagues belonging to the Department of Informatics, Faculty of Engineering, Yamagata University who gave me useful information about Computer Programming IV.

References

1. JABEE web page http://www.jabee.org/.