

효과적인 프로그래밍언어 교육에 대한 연구

A Study on Effective Education of Programming Language

조 재 수*

Jae-Soo Cho*

요 약

본 논문에서는 학생들이 프로그래밍언어를 더 잘 이해하고, 프로그래밍 능력을 향상시킬 수 있는 몇 가지 방법을 제안한다. 현재 컴퓨터공학부뿐만 아니라, 전자공학, 메카트로닉스공학 등에서도 프로그래밍교육이 이루어지고 있으며, 특히 컴퓨터공학의 경우 프로그래밍 교과목은 가장 기본적인 교과목이라고 할 수 있다. 하지만, 컴퓨터 프로그래밍 언어를 이해하고, 주어진 프로그래밍 문제를 해석하여 프로그래밍언어로 구현하는 것에 많은 학생들이 어려움을 겪고, 프로그래밍언어를 가르치는 교수자들도 효과적인 프로그래밍 언어 교육에 관심이 많다. 학생들이 좀 더 잘 프로그래밍언어를 이해하고, 프로그래밍 기술을 향상시킬 수 있는 교육방법을 몇 가지 제안하면 다음과 같다. 첫째 학생들이 프로그래밍언어를 잘 이해하기 위해서는 먼저 컴퓨터 프로그램의 기본적인 동작원리 또는 컴퓨터구조에 대한 사전지식이 필요하다. 둘째 문법위주의 주입식 이론교육보다는 다양한 프로그래밍 예제실행 위주의 프로그래밍언어 교육이 더 효과적이다. 셋째 프로그램의 실행과정과 메모리 할당/해제 흐름에 대한 메모리상태 가시화교육이 필요하다. 넷째 효과적인 실습과 과제 프로그래밍문제들을 통한 프로그래밍 응용능력을 배양하도록 하자. 다섯째 본인 스스로 프로그래밍할 수 있도록 유도하고, 프로그래밍의 자유도를 충분히 부여하자. 마지막으로 팀프로젝트를 통한 개인의 프로그래밍 능력을 극대화하는 것이 필요하다.

Key Words : Programing language, Operation principle of computer program, Computer architecture

ABSTRACT

This paper proposes several methods for effective teaching of programing language(especially C). The programing languages are very important and basic subjects in the Department of Computer Engineering and Science. Many students have some difficulty in learning the programming languages. So, we will suggest several effective methods for effective teaching of programming languages. First, in order to understand the programming language effectively, we need to teach some basic computer architecture and operation principles of computer program to students. Second, it is better to teach many example-based programing skills instead of conventional grammar teaching. Third, if we teach the memory status(memory allocation/release) during the program operation, it will be much better to understand the pointer variables and memory problems. Fourth, let's improve the programming skills through effective homework assignments and experiments. Fifth, let students be guided so that they do programing by themselves. Finally, we can maximize the programming skill through the final termproject.

* 한국기술교육대학교 컴퓨터공학부(jaesoo27@kut.ac.kr)

제1저자 (First Author) : 조재수

교신저자 : 조재수

접수일자 : 2010년 10월 18일

수정일자 : 2010년 11월 17일

확정일자 : 2010년 12월 3일

I. 서론

현재 컴퓨터분야에서 사용하고 있는 프로그래밍 언어는 매우 다양하다. C/C++ 언어를 비롯해서, Basic, Java, C#, PHP 등 매우 다양하고, 각각의 프로그래밍 언어가 가지고 있는 특징도 다르다. 컴퓨터 관련 전공 학생 또는 다른 전공의 학생들이 처음 프로그래밍언어(특히 C언어) 교과목을 수강하게 되면서 많은 학생들이 프로그래밍언어에 대한 어려움을 가지고 있는 것이 현실이다. 컴퓨터공학부의 경우 4년 내내 프로그래밍언어교과목이 아닌 다른 교과목에서도 이러한 프로그래밍언어를 이용하여 수많은 과제와 프로젝트를 수행하게 되는데 프로그래밍언어를 잘 모르는 학생은 대학 4년 내내 전공수업을 제대로 소화하지 못하게 되는 경우가 있고, 심지어 중도에 포기하고 컴퓨터전공을 꺼려하는 현상도 발생한다.

대학 1학년 학생들의 경우 프로그래밍언어 뿐만 아니라 기타 다른 수학 교과목 등도 어려움을 겪는 것이 사실이다. 그 이유 중의 하나는 초등학교에서부터 이어져온 선행학습과 학원교육으로 인한 공교육(대학수업도 일종의 공교육이라고 생각함)에 대한 학생들의 수업태도에도 문제가 많은 것이 현실이다. 즉 1학년 1학기 학생들의 수업은 매우 산만하고, 학생들이 제대로 집중하지 않아 교수자가 수업하기에 매우 힘든 경우가 많다.

본 저자는 컴퓨터공학부에서 다년간 대학 1~2학년들에게 프로그래밍언어(C, C++)를 교육하면서 나름대로 느끼고, 실제 교육을 통하여 경험한 내용을 바탕으로 좀 더 효과적으로 학생들에게 프로그래밍언어를 교육하는 방법을 몇 가지 제안하고자 한다. 특히 다양한 프로그래밍언어 중에서 C언어 교육에 대한 것이고, 그 이유는 아직도 많은 학교와 산업현장에서 C언어를 많이 사용하고 있고, 가장 기본이 되는 언어라고 생각하기 때문이다.

본 논문의 구성은 2장에서 효과적인 C언어 교육을 위한 기본적인 제안들을 기술하고, 3장에서 결론을 맺고, 몇가지 문제점들을 토론하였다.

II. 효과적인 프로그래밍언어 교육을 위한 제안

본 저자는 다년간의 프로그래밍언어 교과목을 담당하면서 다음과 같이 나름대로 효과적인 프로그래밍언어(C언어) 교육에 대한 방법을 제안한다.

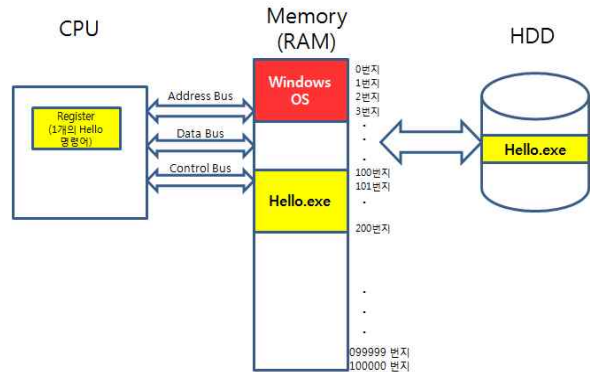


그림 1. Hello.exe 프로그램이 컴퓨터에서 실행되는 과정
Fig.1. The execution process of Hello.exe program

1. 컴퓨터 프로그램의 기본적인 동작원리 및 컴퓨터구조에 대한 사전교육

프로그래밍언어를 학습함에 있어서 먼저 간단하게나마 컴퓨터 프로그램이 어떻게 동작하는지 그리고 컴퓨터 구조가 어떻게 되는지에 대한 사전지식은 학생들이 프로그래밍언어를 이해하는데 매우 도움이 된다. 이러한 컴퓨터 프로그램의 동작원리와 컴퓨터 구조에 대한 사전지식을 가지고 있는 학생과 그렇지 못한 학생들이 프로그래밍언어에 대한 이해도적인 측면에서 현저하게 차이가 날 수 있다.

학생들이 C언어를 학습하면서 가장 어려워 하는 것 중의 하나가 포인터(pointer)이며, 포인터의 포인터 즉 이중포인터의 개념 및 사용은 더욱더 어려워 한다. 이러한 포인터변수(즉 다른 변수의 주소값을 저장하는 변수)를 잘 이해하기 위해서는 [그림 1]과 같이 HDD에 저장되어 있는 Hello.exe라는 실행파일이 어떠한 절차로 컴퓨터에서 실행되는지에 대한 프로그램의 동작원리를 먼저 이해하는 것은 프로그램언어를 학습하는데 있어서 매우 필수적이라고 생각한다. 즉 컴퓨터의 구조(메모리, 주소값 등) 또는 컴퓨터 프로그램이 실행되는 동작원리(HDD에 저장되어 있는 실행가능한 프로그램이 메모리로 옮겨지고, 메모리에 있는 명령어 하나하나가 차례차례 CPU 내의 레지스터(register)에 전달되어 명령어가 해석된 후, 실행되는 절차)를 간단하게 교육시킬 필요가 있다. 이때 메모리의 주소와 주소버스, 데이터버스 등의 역할 등을 알고 있다면, 포인터(다른 변수의 주소값을 가지고 있는 변수)를 이해하는 데 많은 도움이 된다.

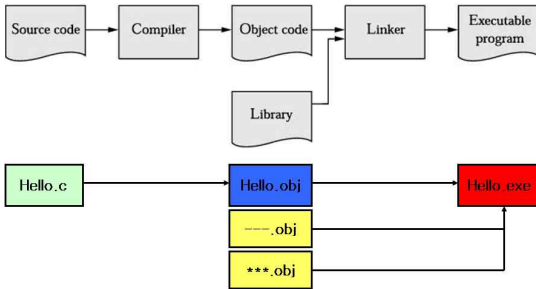


그림 2. 소스코드에서 실행 가능한 프로그램까지의 과정
Fig.2. Programming process from the source code to execution program

참고로 컴퓨터구조(computer architecture)라는 교과목에서 이러한 컴퓨터의 전체적인 구조와 동작원리 등을 교육하고 있지만, 아쉽게도 컴퓨터구조 교과목은 2학년에 개설되어 있는 경우가 많다. 본 저자가 주장하는 것은 컴퓨터구조 교과목을 먼저 가르치자고 주장하는 것은 아니고, C언어를 좀 더 효과적으로 교육하기 위해서 꼭 필요하다고 생각되는 기본적인 컴퓨터의 구조와 컴퓨터에서 프로그램이 실행되는 과정을 학생들에게 꼭 교육시킬 필요가 있다고 생각한다.

그리고 당연한 내용이지만, [그림 2]와 같이 프로그래머가 작성하는 텍스트기반의 소스 프로그램(Hello.c)이 컴파일러(compiler)라는 전문 소프트웨어에 의하여 목적파일(Object file, Hello.obj)로 컴파일되고, 링커(linker)라는 또 다른 전문 소프트웨어에 의하여 다양한 라이브러리(library) 함수와 프로그래머가 작성한 목적파일을 결합시켜 하나의 최종 실행 파일(Hello.exe)을 생성시켜주는 과정을 분명하게 이해시킬 필요가 있다.

프로그래밍언어를 시작하기 전 또는 프로그래밍언어를 교육하면서 이러한 내용들을 부분 부분적으로 학생들에게 교육한다면 학생들이 프로그래밍언어를 좀 더 잘 이해하고, 적용하는데 더 효과적이다.

2. 문법위주의 주입식 교육보다는 다양한 프로그래밍 예제실행 위주의 교육

프로그래밍언어 교육에 있어서 어느 정도는 C언어의 다양한 문법 교육이 필요하다. 하지만 이러한 문법 즉 사용법은 학생들이 다양한 예제들을 직접 실행해 봄으로써 문법적인 것도 익히고, 그 문법이 실제 프로그래밍에서 어떻게 사용되고 있는지를 확인하는 것은 학습자 입장에서 매우 효과적이다. 처음 프로그래밍을 학습하는 학습자 입장에서는 매우 다양한 C언어 사용 예를 다양한 예제를 통하여 익힌다

면 향후 본인이 작성해야 할 본인의 프로그램에서 그러한 예들을 그대로 활용할 수 있기 때문이다.

처음 배우는 입장에서는 다른 사람이 작성한 많은 프로그램을 실행해 보고, 사용방법들을 익히는 것이 향후 본인의 프로그램에 응용하는 것에 매우 효과적이다. 예를 들어 C언어 “함수”를 교육할 때 함수에 대한 기본적인 문법(사용법)을 교육한 후, 다양한 역할을 하는 함수들에 대한 예제들을 직접 실행해 봄으로써 함수를 학습하는 것이 훨씬 효과적이다. ‘포인터’ 부분을 교육할 때도 ‘포인터’에 대한 기본적인 내용을 교육한 후, 다양한 예제들을 실행해 봄으로써 포인터가 무엇이고, 왜 포인터를 사용해야만 하는지, 포인터를 사용해야만 하는 다양한 예제를 학생들에게 직접 실행해 보게 함으로써 포인터를 더 잘 이해하고, 그 사용법을 좀 더 효과적으로 익힐 수 있다.

참고로 이때 다양한 예제들을 학생들이 실행해 볼때 코드 하나하나를 직접 학생들이 타이핑하면서 해 보면 좋겠지만, 수업시간이 한정되어 있고, 다양한 예제들을 많이 실행해 보기 위해서는 예제코드들은 학생들에게 미리 직접제공한 후, 그 예제들을 바로 복사해서 실행한다면 시간적인 문제를 해결할 수 있다.

3. 프로그램 실행과정에서의 메모리 할당/해제와 같은 메모리 상태 가시화 교육

학생들이 프로그래밍언어를 배우면서 가장 어려워하고, 힘들어하는 부분이 메모리 관련 포인터변수 사용이고, 가장 실수를 많이 하는 부분이 메모리 관련 내용이다. 이러한 포인터관련 내용을 잘 이해하고 프로그래밍하기 위해서는 실제 프로그램이 컴파일되는 과정(compile time)과 실행되는 과정(run time)에서 메모리적인 측면 즉 메모리가 언제 어떻게 할당되고 해제되는지에 대해 명확하게 학생들에게 이해시킬 필요가 있다. 이를 위해서는 C언어의 메모리 구조(그림 3)와 메모리공간이 할당되는 시기, 누가(OS 또는 프로그래머) 메모리를 할당하는 것인지 그리고 메모리의 용도 등에 대한 이해가 필요하다.

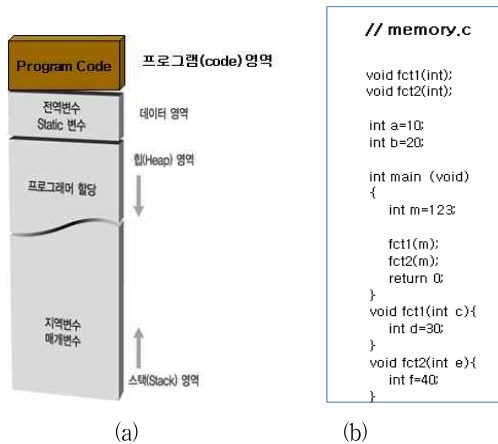


그림 3. (a) C언어에서의 메모리 관리 구조, (b) 간단한 프로그램 예

Fig. 3. (a) Memory management structure in C-language, (b) an example of simple program

간단한 예로[1] 그림 3(a)와 같은 C언어에서의 메모리 관리 구조를 이해하고, 그림 3(b)에서 예시한 것과 같이 간단한 프로그램(memory.c)을 컴파일 한 후, 프로그램이 시작되기 전의 메모리 구조(그림 4(a))와, 프로그램이 시작된 후(즉 main 함수가 호출된 후, 그림 4(b))의 메모리 내용, main 함수내의 함수들이 하나하나 호출되고, 실행된 후에 각각의 함수에서 선언된 다양한 지역변수 및 매개변수들이 메모리에 언제 어떻게 할당되고 해제 되는지에 대한 흐름을 학생들에게 명확하게 이해시키는 것은 매우 중요하다. 학생들이 C언어에서 변수선언과 메모리 할당/해제 등에 대한 관계를 이해하는 데 매우 중요하기 때문이다.

간단한 프로그램이 실제 실행될 때 각 변수와 메모리 등이 언제 어디에 할당되고 언제 해제되는지에 대한 개념을 이해한다면 C언어에서 메모리에 대한 내용과 포인터를 더 잘 이해할 수 있을 것이다.

4. 실습과 과제를 통한 프로그래밍 응용능력 배양

프로그래밍언어 이론시간에 배운 문법을 다양한 응용사례 예제를 통하여 사용법을 학습한 학생들에게 이제 본인이 직접 이론수업시간에 배운 내용을 실습시간에 프로그래밍해 보는 과정이 필요하다. 이때의 실습내용은 수업시간에 배운 문법과 다양한 예제들을 조금만 활용하고 응용하면 프로그래밍할 수 있는 실습문제를 학생들에게 제시한다. 본 저자의 경우 일반적으로 두 문제를 실습문제로 학생들에게 제시하며, 두 문제 중 한 문제는 비교적 쉬운 내용이고

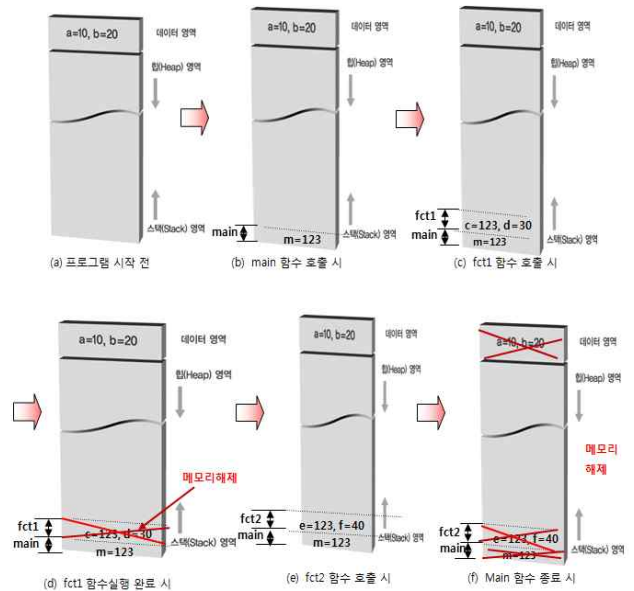


그림 4. 간단한 프로그램(memory.c)의 실행과 메모리 할당/해제의 흐름

Fig. 4. Memory allocation/release process for a simple program

다른 한 문제는 어느 정도 시간과 노력을 필요로 하는 문제를 학생들에게 제시한다. 그 이유는 학생들의 수준차이가 심한 편이어서 쉬운 문제는 프로그래밍 수준이 낮은 학생도 어느 정도 이론 수업시간의 내용을 이해하면 프로그래밍할 수 있도록 함이고, 비교적 어려운 문제는 수준이 높은 학생도 도전적으로 흥미를 가질 수 있도록 함이다. 이때 수준이 낮은 학생의 경우 경우에 따라서는 실습시간이 부족하여 그 날 저녁까지 실습을 하게 되는 경우가 있으며 이러한 경우 교과목 실습홈페이지(예, <http://psl.kut.ac.kr>)에 실습결과를 집에서도 업로드(upload)하게 함으로써 프로그래밍의 실습 효과를 높이도록 하였다. 즉 시간이 부족하여 본인 스스로 생각하고, 코드화하는 과정을 생략하고, 남의 생각과 코드를 그대로 카피해서 제출하지 않도록 하는 것은 프로그래밍 응용력을 높이기 위해 매우 중요한 과정이다.

실습과 과제를 체점하고 실습수업시간에 대한 조교 및 교수의 역할도 매우 중요한데, 개인적으로 학생들이 조교와 교수의 도움 없이 스스로 생각하고, 스스로 프로그래밍 할 수 있는 분위기를 조성해 주는 것이 매우 중요하다고 생각한다. 즉 프로그래밍 실습시간에 한사람 한사람의 코드를 직접 조교나 교수가 옆에서 지도해주기 보다는 학생들이 스스로 예

리를 수정하고, 코드에 한줄이라도 스스로의 힘으로 코드화할 수 있도록 도와주는 것이 필요하다. 예를 들어 학생에 따라서는 프로그램을 하고 컴파일 한 후 에러가 발생하면 그 에러메시지를 확인하지도 않고, 그냥 조교나 교수에게 에러가 왜 났는지를 모르겠다고, 에러를 수정해 달라고 하는 경우가 있다. 이러한 경우 조교나 교수는 그 에러를 직접수정해 주기 보다는 학생스스로가 에러메시지를 해석해서 그 에러가 어떤 에러인지를 상의하고, 그 에러 메시지가 무엇을 말하는 것이므로 코드에서 본인이 그것을 찾아내어서 수정할 수 있도록 도와주어야지 직접 그 에러를 수정해 주어서는 그 학생의 프로그래밍 능력은 향상되지 않는다고 생각한다.

그리고 실습과 과제를 채점하고, 조교는 가장 잘 프로그램한 샘플코드(Best Code)를 학생들에게 제시해 줌으로써 학생들이 자신의 코드와 Best Code를 비교함으로써 자신의 코드의 문제점들을 스스로 수정할 수 있도록 한다. 그리고 조교는 학생들이 반복적으로 실수하거나 잘못된 코드에 대하여 교수와 학생들에게 피드백해 주는 과정이 반드시 필요하다.

5. 본인 스스로 프로그램할 수 있도록 유도하고, 프로그래밍의 자유도를 부여하자.

프로그래밍 능력을 배양하기 위해서는 한 줄의 코드도 본인 스스로 할 수 있는 습관을 들이는 것이 매우 중요하다. 하지만 프로그램의 특성상 많은 학생들이 실습과 과제를 다른 사람의 것을 그대로 카피(copy)하는 경우가 많고, 이러한 것을 처음 실습과 과제채점에 반영하지 않으면 많은 학생들이 한 학기 내내 다른 사람의 코드를 그대로 복사해서 제출하는 경우가 많다. 이러한 것을 방지하기 위해 처음 1~2주 동안은 반드시 다른 사람의 코드를 카피해서 제출한 사람과 그것을 그대로 보여준 두 사람 모두 실습과 과제점수에 불이익(0점 처리)을 줌으로써 학생들이 본인 스스로 실습과 과제를 수행할 수 있도록 최대한 조교와 교수가 노력을 기울이는 것이 매우 중요하다.

그리고 프로그래밍의 자유도를 부여하는 것도 매우 중요하다. 프로그램의 자유도란 프로그램이라는 것이 수학문제를 푸는 것처럼 1개의 정답이 있는 것이 아니고 사람에 따라서 똑같은 Input/Output 을 가지는 프로그램이라도 그 구현방법은 매우 다양할 수 있기 때문이다(물론 때로는 그 수업에서 배운 내용을 반드시 사용하게 하는 등의 제약조건도 필요하다).

6. 팀프로젝트를 통한 개개인의 프로그래밍 능력 극대화

개별적으로 1년동안 배운 다양한 프로그래밍언어 요소들을 개인적인 팀프로젝트를 통하여 각 개인의 프로그래밍 능력을 극대화하도록 한다. 그림 5는 학생들에게 제시한 팀프로젝트 안내서의 한 예[2]이다. 크게 4가지 주제에 대하여 교수가 학생들의 수준에 따라 각 주제에 학생들을 고르게 배정하였으며, 팀프로젝트는 팀단위가 아닌 개인 프로젝트로 진행하는 것이 더 효과적이다. 학생들을 각 주제에 배정할 때 학기 중의 실습과 과제평가등을 통하여 학생들의 수준을 미리 파악한 후, 학생들의 수준을 고려하여 고르게 배정될 수 있도록 할 필요가 있다. 여기서 고르게 배정된다는 한 주제에 프로그램을 잘 하는 학생들 또는 못하는 학생들이 몰리지 않도록 하는 것을 말한다. 그 이유는 비록 개인프로젝트로 진행하더라도 같은 주제를 가진 학생들은 때에 따라서는 서로 상의하고, 토의해서 하는 경우가 많으므로 각 주제에 대하여 프로그래밍을 잘 하는 학생과 그렇지 못한 학생들을 고르게 배정해주는 배려가 필요하다.

팀프로젝트의 주제는 1학년때 C언어에서 한 주제와 2학년 C++언어에서 수행하게 될 팀프로젝트와 같은 주제로 하게 함으로써 C언어로 작성한 프로그램을 객체지향언어인 C++언어로 2학년때 프로그래밍 함으로써 C언어와 C++언어의 차이를 느낄 수 있도록 하였다.

팀프로젝트 안내서 예제에 보면 알겠지만, 공학인종에서 이야기하는 설계제한요소 등에 대한 내용은 포함하지 않고 있다. 그 이유는 학생들의 능력이 다양하고, 오히려 설계제한요소를 통한 교육효과보다는 개개인의 프로그래밍 능력을 개인에 따라서 스스로 결정하고, 스스로 기능을 선택하라는 의미가 더 크다고 할 수 있다. 똑 같은 주제로 프로그래밍 하는 팀 프로젝트이지만, 개인의 프로그래밍 능력에 따라서 최종적인 결과는 매우 다르다. 학생들의 수준에 따라서 그 기능을 매우 다양하게 구현하기 때문에 실습과 과제는 50줄 내외의 프로그램을 구현해 본다면, 학기말 팀프로젝트는 작게는 250라인에서 많게는 1500 ~ 2000 라인의 프로그램을 작성하는 학생도 있다. 비록 프로그램의 라인수가 많다고 해서 완성도가 높은 것은 아니지만, 처음 프로그램을 배우는 1학년의 입장에서 프로그래밍 라인수가 많다는 것은 더 많은 기능을 포함하고 있다고 판단할 수 있다. 이러한 과정에서 중요한 것은 그 주제에 대한 기능의 Li

**2010년도 2학기 C프로그래밍II
Term Project 안내서**

- 조제 교수: 조재수

1. 팀프로젝트 주제 (개인프로젝트)

(1) 주제 1: 비디오 삽 판리 프로그램

- 김동진,최윤희, 김준우,기철민,이희덕,정승욱,김지훈,박지민,박한

(2) 주제 2: 도서 판리 프로그램

- 김덕균,조준성,김다예,최소라,황동원,표경수,허정은,박종희,황지형

(3) 주제 3: 학생관리 프로그램(성적관리 포함)

- 윤나영, 김효영,박희윤,이현수,김제환,주승기,김진학,한준희,이현수

(4) 주제 4: 주차관리 프로그램

- 김웅태, 김다연,조지은,김유라,이제원,한희종,박은승,박성균

2. Term Project 수행 시 유의 사항

2.1 Term Project 제안서 제출

- 12월 6일(월) 저녁 12시까지 과제계좌 Homepage에 제출

2.2 Term Project 결과물(CD) 제출내용

- 최종 프로그램
- Term Project 최종보고서(HWP 또는 WORD)
- 최종 발표자료(PPT)

(Term Project 보고서: HWP 또는 Word, 이러한 문서란 처음 프로그램은 작성 시 클래스 구조에서부터 멤버함수, 멤버변수 설명 등을 자세하게 기록하여야 합니다. 본인의 프로그램을 다른 사람이 이 문서만 봤을 때 전체 프로그램 구조를 한 눈에 파악할 수 있도록 작성하도록

한다.)

3. Term Project Grading

- 1) 프로그램 완성도: 50%
- 2) Term Project 제안서 및 최종 보고서 내용: 30%
- 3) 최종 발표자료 및 발표: 20%

4. Term Project 최종발표

12월 27일(월요일) 오전 10시~

5. 주의사항

- 만약 Term Project의 최종 결과물이 다른 사람(선배나 동배)의 코드를 그대로 Copy한 경우는 학점이 0가 될 예정이므로, 반드시 본인의 노력으로 Term Project를 완성해서 제출하시기 바랍니다. (참고로 다른 사람의 코드나 다른 사람 것을 일부 참고하였을 경우는 반드시 그 출처를 보고서에 명시하시기 바랍니다.)

이상입니다.

그림 5. 팀프로젝트 안내서 예시

Fig. 5. An example of termproject guideline

-mit를 주지 않고, 개개인의 능력에 따라서 기능을 개인이 스스로 선택하게 하는 것이다.

팀프로젝트 기간은 약 0.5개월에서 1.5개월 정도 되고, 팀프로젝트의 최종발표는 모든 학생이 발표하여야 하지만, 시간적 제약 및 방학 등을 고려하여 각 주제에 대하여 대표적인 1~2명이 최종발표를 하게 한다. 이때 그 1~2명은 전체적으로 발표를 희망하는 학생들의 지원자를 먼저 받고, 지원자가 없는 경우 팀 프로젝트를 가장 잘 수행한 학생을 대표로 발표하게 함으로써 다른 학생들이 자신이 작성한 프로그램과 비교할 수 있도록 하면 교육효과가 더 있는 것 같다.

III. 결론 및 토의

본 논문에서는 나름대로 1~2학년들에게 프로그래

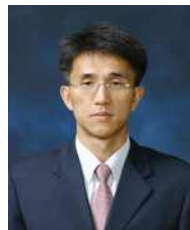
밍언어(C, C++)를 교육하면서 경험한 사례를 바탕으로 좀 더 효과적인 프로그래밍 교육에 대한 방법을 제안하였다. 하지만, 제안한 방법을 교육현장에 잘 적용하기 위해서는 다양한 추가적인 노력이 필요하다. 예를 들어 프로그래밍언어 문법만의 교육이 아닌 다양한 예제중심의 응용사례 중심 교육을 하기 위해서는 산업현장에서 실제 응용되고 있는 많은 예제들을 찾고 응용할 필요가 있다. 그리고 팀프로젝트를 하면 안하는 것보다는 더 효과적이라는 것은 알고 있지만, 학기가 종료된 후 1~2주 후에 기숙사도 종료되는 상황에서 팀프로젝트 발표를 위해서 학생들을 등교시켜야 하는 등 학생들의 불만들이 있는 것이 사실이다. 계속해서 더 좋은 프로그래밍언어 교육을 위해서는 많은 프로그래밍 예제들을 발굴하고, 다양한 현장에서 프로그래밍언어들이 어떻게 응용되고 있는지를 연구하는 것이 필요하며, 좀 더 많은 시간을 할애해서 학생들의 실습과 과제 및 팀프로젝트 결과를 학생들에게 피드백해 주는 것이 필요한 것 같다. 그리고 이러한 교육방법에 대한 효과검증은 아직 연구가 되지 않은 상황이다. 이러한 교육방법에 대한 효과를 검증하기 위해서 추후과제로 학생들의 수업반응도 응답이나 추가적인 설문조사 등을 통하여 그 효과를 분석할 필요가 있다.

참고 문헌

- [1] “C프로그래밍”, 윤성우 저, FREELEC
- [2] 조재수, “효과적인 프로그래밍 교육에 대한 연구”, 한국실천공학회 2010추계학술대회

조재수 (Jae-Soo Cho)

정희원



1993년 2월 : 경북대학교 전자공학과 공학사
 1996년 2월 : 한국과학기술원 (KAIST) 전기및전자공학과 석사
 2001년 2월 : 한국과학기술원 (KAIST) 전자전산학과 공학박사
 2001년 3월~2003년 8월 : (주)삼성전자, 책임연구원
 2003년 9월~현재: 한국기술교육대학교 컴퓨터공학부 부교수

<관심분야> Visual Surveillance, Automatic Video Tracking, Pattern Recognition, Machine Vision