

프로그래밍 언어 교육현황과 개선제안

Present State of Programming Language Education and Suggestions for Its Improvement

임 재 열*

Jae-Yeol RHEEM*

요 약

IT기술의 발달로 SW 중요성이 높아짐에 따라서 대학졸업자들의 프로그래밍 능력에 대한 산업체 요구가 점점 높아지고 있다. 그 결과로, 시스템 위주의 전자공학 전공 교과목에서도 프로그래밍이 이미 많은 비중을 차지하고 있으며, 컴퓨터 실습실과 같은 교육환경은 상당히 개선되었고, 교수들이 강의의 질 및 교육 결과를 향상시키고자 많은 노력을 하고 있다. 하지만, 여전히 프로그래밍 교육에 대한 사회 및 산업체의 요구를 만족시키지 못하는 것도 현실이다.

본 논문에서는 한국기술교육대학교 전기전자통신공학부 전자전공에서 이루어지고 있는 프로그래밍 교육 현황을 소개하고, 프로그래밍 교육의 근간이 되는 1학년 대상 프로그래밍 언어 교육에 대한 문제점을 학생, 교재, 담당 교수, 수강학생수 그리고 수업시간 면에서 고찰 한 후, 개선방안을 제안한다.

Key Words : C Language, Programming Language Education, Active Learning

ABSTRACT

As IT technology improves, software is being more and more important. Industry demands higher level of programming ability from university graduates. As a result, programming related courses already take higher possession of electronics department curricula. But even though the environment likewise computer laboratory is improved as much, and professors have tried to enhance the quality of their lectures and educational results, still programming education does not match the demand of our society and industry.

In this paper, we introduce the overview of programming education in electronics department, KUT and discuss problems in programming language education especially for the 1st year students, since programming education to the 1st year university students becomes one of the most important part as it deals with fundamentals on programming language and we suggest its improvement directions in regards with students, teaching materials, professors, class size, and class hours.

* 한국기술교육대학교 전기전자통신공학부(rheem@kut.ac.kr)

제1저자 (First Author) : 임재열

교신저자 : 임재열

접수일자 : 2011년 4월 29일

수정일자 : 2011년 5월 27일

확정일자 : 2011년 6월 06일

I. 서론

IT기술의 발달과 더불어 소프트웨어의 중요성이 점점 더 강조되고 있으며, 대학 졸업자에게 산업체에서는 글로벌 능력과 함께 창의적 문제해결 능력을 요구하고 있어, 대학교육은 학생수 감소라는 물리적 위기 속에 질적 위기를 맞고 있는 중이다.

소프트웨어의 중요성은 이미 20세기 후반부터 강조되어 하드웨어에서 소프트웨어로 패러다임이 바뀌었다. 그러한 결과 시스템 위주의 전자공학 전공 교과목에서도 프로그래밍이 이미 많은 비중을 차지하고 있고, 그 근간이 되는 1학년 대상 프로그래밍 언어 교육은 점점 더 중요해 지고 있다.

그리고 산업체의 요구수준에 맞추어 교수들은 프로그래밍 교육의 질을 높이고자 부단히 노력해 왔으며 [1][2], 학부 및 학과에서는 실험실습 및 기자재 개선을 통하여 프로그래밍 교육환경을 지속적으로 향상시켜 왔다 [3]. 그러나, 졸업생들의 설문조사 결과, 2008년 이후 졸업생들이 지속적으로 중급 이상의 프로그래밍 능력을 갖추어야 한다는 지적을 계속하고 있는 것으로 보아 [3], 그간의 노력에도 불구하고, 사회 및 산업체의 요구수준을 따라가지 못 하고 있는 것도 주지하고 있는 사실이다. 아울러, 프로그래밍의 중요성 및 체계적인 교육을 고려하면, 전자공학에서의 프로그래밍에 대한 교육은, 컴퓨터 및 소프트웨어를 주전공으로 하는 학부 및 과의 체계적이고 다양한 언어 교육 및 시도를 따라가지 못 하고 있는 것도 엄연한 실정이다.

현재 많은 대학에서 1학년 프로그래밍 언어 교육이 C++에서 C# 등으로 전환되고 있지만, 전자공학 분야에서는 마이크로프로세서 등 하드웨어 기반의 프로그래밍이 많이 요구되어 1학년에서는 특히 C언어가 주 언어로 교육되고 있다. 점차 소프트웨어 비중이 커질 것에 대비하여 데이터구조 및 알고리즘, 소프트웨어 공학 등 체계적인 프로그래밍 교육과 이 러닝 학습 등 학생들의 다양한 학습지원이 이루어져야 할 것으로 보인다 [4]-[6].

본 논문에서는 한국기술교육대학교(KUT: Korea University of Technology and Education) 전기전자통신공학부 전자전공에서 이루어지고 있는 프로그래밍 관련 교과목 운영 현황 및 문제점을 소개하고 개선방안을 제시하는 것을 목적으로 한다.

본 논문의 구성은 다음과 같다. II장에서는 프로그

래밍 관련 교과목과 사용언어, 1학년 C언어 교육 현황 등 프로그래밍 교육 현황을 살펴보고, III장에서는 프로그래밍 교육의 문제점 및 개선방안을 고찰한다. 그리고 IV장에서 결론을 맺는다.

II. 프로그래밍 교육 현황

1. 프로그래밍 관련 교과과정

한국기술교육대학교 전기전자통신공학부 전자전공에서 프로그래밍 언어를 활용하는 넓은 의미의 프로그래밍 관련 교육 현황은 표 1과 같다. 표 1에 있는 과목들은 모두 실습시간을 포함하고 있어, 실질적으로 프로그래밍 실습이 병행하여 진행되고 있는 교과목 들이다.

표 1. 프로그래밍 교육관련 교과목 및 사용언어
Table 1. Programming related courses and languages in use

학 기	교과목명	사용언어	비고
1-1	● 프로그래밍 기초	C	필수
1-2	● 프로그래밍 언어	C	필수
2-2	● 마이크로프로세서 I ● 수치해석 및 실습 ● 디지털시스템 설계 및 실습	Assembler	필수
		MATLAB	필수
		C	필수
3-1	● 마이크로프로세서 II ● 임베디드 SOC 설계 및 실습	C	선택
		C	선택
3-2	● 디지털 신호처리 및 실습 ● 윈도우 프로그래밍 ● 임베디드 소프트웨어 및 실습	MATLAB	필수
		VC++	선택
		C (Linux)	선택
4-1	● 화상처리 및 실습 ● DSP 하드웨어 및 실습	VC++	선택
		C	선택

표 1에서 볼 수 있듯이, 전자전공 4년에 걸친 교육에서 프로그래밍이 차지하는 비중은 매우 크다는 것을 알 수 있다. 아울러, 전자회로 설계 등 하드웨어 관련 교과목도 전용 시뮬레이터 등의 실습을 통하여 진행되는 것이 많아, 소프트웨어의 사용 및 이에 따른 프로그래밍의 비중은 점점 높아지고 있는 추세이다.

프로그래밍 언어 자체에 대한 교육은 1학년에서만 이루어지며, 2학년 이후에는 모두 관련 교과목의 학

문적 이해를 돕기 위한 실습과정에서의 프로그래밍이 이루어지기 때문에 프로그래밍의 기초를 닦는 1학년 C언어 프로그래밍의 마스터는 점점 더 중요해지고 있다.

표 1에서 알 수 있듯이 프로그래밍 관련 교과목은 모두 실습을 병행하고 있으며, 이수기준은 3학점 2시간 이론 2시간 실습으로 주당 4시간의 수업이 16주에 걸쳐 이루어지고 있다. 2009년까지는 2학년 1학기에 “윈도우 프로그래밍”과목이 있어서 매 학기 1과목의 프로그래밍 관련 과목이 순차적으로 있었으나, 2010년 교과과정 개편시 3학년 2학기로 현재와 같이 이동하였다.

프로그래밍관련 부족한 부분에 대한 보충은 학기 중에도 일부 진행되는 경우도 있으나, 일반적으로 외부 전문가 초청 세미나나 단기 특강 또는 대학원 석/박사과정 학생들이 활용한 단기 프로그램 형태로 주로 방학 중에 이루어져, 학부생들의 프로그래밍 능력 부족한 부분을 보충하고 있다.

2. C언어 교육 현황

가장 기초가 되는 C언어 교육은 “프로그래밍 기초”, “프로그래밍 언어” 2개의 교과목으로 1학년 1, 2학기에 걸쳐서 전공필수로 진행된다. 개교 초기부터 학생들의 이해 수준 및 제대로 된 실습교육을 위해서 2학기에 걸쳐서 수업을 하고 있으며, 3학점 이론 수업 2시간 실습 2시간 구성으로 주당 4시간 수업이 진행된다.

2010년까지는 전기/전자/정보통신 3전공 신입생과 복학학생 및 재수강자를 위하여 1분반을 추가 개설하여 전공당 입학정원 45명 기준으로 4분반을 개설하였으나, 2011년에는 교육효과를 고려하여, 6개 분반을 개설하고 있으며, 한 분반은 30명 전후로 운영하여 교육의 질적 향상을 꾀하고 있다. 특이한 점은 이중 2개 분반은 원어민 전공전임교수가 담당하고 영어로 수업을 진행하고 있다. 담당교수는 2010년까지는 주임교수 1인 및 강사로 구성 운영 되었지만, 2011년부터는 1분반만 강사가 담당하고 나머지는 모두 전임 교수가 맡아 이점에서도 교육 내실을 기하고 있다. 실습조교는 지원되고 있으나, 이론 실습을 구분하기보다는 대부분의 교수들이 이론 및 실습을 병행하고 있어, 실습시 조교들에 대한 의존도는 매우 약하다고 볼 수 있다. 2008년까지는 실습교육의 수월성 및 강사고려로 이론 및 실습이 4시간 연강이 주로 이루어졌으나, 2009년 이후 학교의 분리운영 방침

에 따라, 주 2회 2시간씩 분리된 시간표를 운영하고 있다. 진도 및 교과내용은 6개 분반이 모두 보조를 맞추고 있으며, 성적처리는 학점분포 비율에 의한 분반별 철저한 상대평가를 하고 있다. 일반적으로 중간고사:기말고사:보고서 및 실습:출결을 30:30:30:10 비율로 반영하고 있다.

강의 진도는 표 2와 같다. 1학기에는 순서도 작성 및 함수 작성에 중점을 두고 있으며, 학기말 1달간은 함수 작성 연습이 충분히 될 수 있도록 유도하고 있다. 2학기 때는 파일처리 및 종합 응용프로그래밍 (Term Project)에 중점을 두고 있으며, 학기말 1달을 응용 프로그램 개발에 할당하고 있다. 학기초 1학기 내용에 대한 복습을 정규적으로 할당하고 있는 것도 특징이다.

표 2. C언어 학기별 주요 내용
Table 2. Typical example curricula of C language for two semesters

학기	월	주요 내용
1	3	기본개념, 순서도, 데이터 타입, 변수, 콘솔 입출력
	4	연산자, 선택문, 순환문
	5	수학함수, 함수 (포인터, 배열)
	6	함수 연습
2	9	1학기 복습, 함수 연습, 포인터
	10	배열, 구조체 및 유니온
	11	파일, 동적메모리 할당, 기타
	12	응용프로그래밍 연습 (Term Project)

1학기 함수 연습은 개별함수 만드는 연습을 통한 프로그래밍 능력 배양과 더불어 입력/처리/출력을 함수로 분리하여 실행하는 연습을 통하여, 함수의 입출력 처리 능력을 키워주고자 노력하고 있으며 이에 필요한 최소한의 포인터 및 배열 내용을 1학기 때 다루며, 2학기 때 파일 처리를 배운 후 파일 처리 및 함수를 개별 파일로 분리하여 컴파일 활용하는 것을 배움으로써 함수 활용에 대한 것을 마무리하고 있다.

2학기 응용 프로그래밍 연습은 학생들의 분석력 및 집중도를 키워주기 위하여 단계적으로 복잡한 입출력이 필요한 프로그램 과제와 로직 개발에 창의력이 필요한 계산 과제를 각각 2-3주에 걸쳐서 개발하여 발표하는 것으로 구성하고 있다. 과제 주제는 같으며, 발표는 모든 학생들이 자신의 과제결과를 발표하고, 학생들도 평가에 참여함으로써, 평가요소, 발표요령, 다른 학생 발표의 경청 방법도 함께 습득할 수 있도록 유도하고 있다.

매주 실습에서 주안으로 삼는 것은 교과서적인 프로그래밍 학습을 지향하여, 프로그래밍 절차에 준한 프로그램 작성이 되도록 학생들에게 강력하게 요구하고 있다. 실습 결과 및 과제 리포트 작성 시 반드시 그림 1과 같은 프로그래밍 절차를 준수하여, 문제 정의 및 분석, 입출력 설계, 순서도, 코딩 및 디버깅, 샘플 테스트, 문서화(Documentation)를 간략하게 기술하게 하고 있다. 순서도(flow chart)에서는 프로그램 코드와 일치, 코딩 및 디버깅에서는 적절한 주석(comment) 달기와, 에러 및 디버깅 기록하기, 문서화에서 반드시 결론을 작성하도록 하고 있다. 참고로, 1학년 과제 리포트는 워드 프로세서 사용 작성을 허용하지 않고, 수기 작성을 원칙으로 하고 있다.

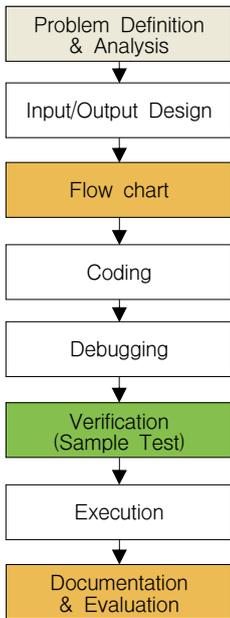


그림 1. 전형적인 프로그래밍 절차 예
Fig. 1. Typical programming procedure example

프로그램 실습 또는 과제상의 전형적인 실수에 대한 검토는 실물 화상기를 이용하여, 리포트 자체를 스크린에 비추어, 직접 수정해 주고 있으며, 과제에 대한 즉각적인 피드백의 결과, 학생들의 주의집중 및 만족도도 매우 높아, 2학기의 경우 학생들이 본인의 과제결과를 검토해 주길 원하는 학생들이 늘고 있다.

그림 2는 C언어 교육 현황결과로써, 과거 7년간의 프로그래밍 기초 교과목 성적의 누적 백분율을 보여 준다. 여기서 x축은 점수대로 점수대 90은 100에서 90점을 의미하며, y축은 누적 백분율(%)이다. 평균

수강학생수는 46명이었으며, 성적은 ABC 등으로 상대적 평가를 하기 전의 100점 만점의 절대평가 점수로 중간 및 기말고사, 과제, 출석점수가 반영되었다.

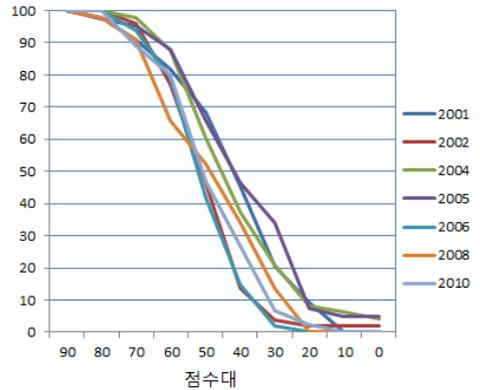


그림 2. 프로그래밍 기초 성적 누적백분율
Fig. 2. Cumulative Percentage of Programming Basics Marks

그림 2를 보면, 50점대를 기준으로 정규분포를 이루고 있음을 알 수 있으며, 년도에 따른 특이특성을 보여주고 있지 않아서, 그동안 발생했다고 믿어지는 학생들의 학력저하에, 교육 노력 등이 상쇄되어 평균적인 특징을 보여주고 있다고 볼 수 있다.

각 점수대별 7년 평균 및 누적평균 백분율을 나타내는 표 3을 보면, 60점대가 가장 많은 25.4%를 차지하고 있으며, 60점 미만의 비율이 54.5%로 과반수를 넘어, 수강생들의 교육성취는 많이 부족함을 알 수 있다. 이는 학생들의 강의평가 만족도가 상대적으로 낮은 것과 일치하고 있어, 이점에 대한 개선이 필요하며, 아울러, 학습부진으로 볼 수 있는 50점 미만 학생이 31.2%, 약 14명에 해당되어, 이 학생들에 대한 대책이 시급함을 알 수 있다.

표 3. 프로그래밍 기초 성적 7년 평균 백분율
Table 3. Average percentage of 7 years Programming Basics Marks

점수대	90	80	70	60	50	40	30	20	10	0
평균 (%)	1.0	5.6	13.5	25.4	23.3	16.6	10.4	2.3	0.3	1.6
누적평균 (%)	100	99.0	93.4	79.9	54.5	31.2	14.5	4.1	1.9	1.6

표 4는 그림 2의 결과를 바탕으로 학생들에게 부여한 학점 비율이다. A와 B를 받은 학생이 56.7%로 과반수를 넘고 있어, 절대점수에 비하여 학점을 상대적으로 잘 주고 있음을 알 수 있다. 반면에 F학점을

받아 학점취득에 실패한 학생은 약 10%인 46명이다. 이는 표 3에서 40점 미만에 해당되며, 이들의 취득 점수를 고려하면, 중간고사 이전에 이미 학습을 포기한 중도포기자로 여겨진다.

표 4. 프로그래밍 기초 평균 학점 비율
Table 4. Average grade percentage of Programming Basics

학점	A	B	C	D	F
평균비율(%)	23.7	33.0	27.1	5.9	9.9

III. 프로그래밍 교육 문제점 및 개선제안

1. 프로그램 교육의 문제점

2011년의 경우, 30명 전후 분반운용 및 전임교수 담당 증가로 인하여, 기존의 문제점들이 많이 해결되었지만, 프로그래밍 교육 특히 1학년 C언어 교육에서 문제는 다음과 같이 요약할 수 있다.

1) 학생: 교차지원으로 인하여 순서도, 수열, 점화식 등에 대해서 학습이 제대로 이루어지지 않은 학생들이 증가하여, 초기적응을 못하여, 관심 및 집중도가 점차 떨어진다.

남학생의 경우 1학년 마치고 군대 가는 학생들이 많고, 이 학생들의 상당수가, 제대 후 본격적으로 공부를 하겠다는 생각을 갖고 있는 학생들이 많아, 초기에 학습에 소홀히 함으로써 프로그래밍 학습에 낙오하는 학생이 발생한다.

2) 교재: 좋은 책은 많이 나와 있으나, 이론 및 실습을 적절 수준에서 진행할 수 있는 워크북 형태의 맞춤형 교재가 없다.

자기주도적 학습력이 떨어지는 현재의 학생들을 위해서는 특히 주기적인 연습을 유도할 수 있는 실습교재가 많은 도움이 될 수 있다.

담당교수: 대학 전체를 보면 1학년 프로그래밍 언어 교육을 맡은 부분 강사에 의존하고 있으며, 학문적 불일치도 문제가 되고 있다. 아울러 프로그래밍 교육에 대한 학문적 경험이 적어, 1학년에 대한 전공에 대한 이해, 프로그래밍의 중요성, 과제 및 피드백 등 학생 맞춤형 지도가 이루어지기가 쉽지 않다.

전공에서 요구하는 이론 및 실습에 균형 잡힌 교수 및 강사를 확보하는 것도 중요한 문제이나, 과거의 관행상 쉽게 누구나 할 수 있는 과목으로 간주하

는 것이 큰 문제이며, 강의에 대한 실질적 평가에 대한 시스템이 갖추어져 있는지도 매우 중요한 문제이나 쉽게 간과되고 있다.

3) 수강학생규모: 효과적인 수업을 이해서는 학생수가 적절해야 한다. 실습의 경우 25명 전후가 한계이며, 선진국의 경우 대개 20인 이하의 실습이 주류를 이루고 있으며, 적을수록 교수 및 학습 효과는 크다. 그러나, 실습을 소규모 분반으로 운영하는 경우 현재의 대학 평균의 교수대 학생비율로는 물리적으로 쉽지 않은 측면이 많다. 학문 특성상 대규모 이론 강의에 소규모 실습 분반운용도 맞지 않으며, 소규모 실습 분반운영이 가능해도 실습 전담 교수 및 전담 교수 확보도 쉽지 않은 현실이다.

4) 수업시간: 현재 한국기술교육대학교는 2시간 이론 2시간 실습 운영을 하고 있으나, 자기적 주도학습에 익숙하지 않은 현재 학생들의 실습시간으로는 절대적으로 부족하다. 수업을 통해서 학습목표를 달성하려면, 최소 3시간 이론, 5시간 실습 시간이 필요할 것으로 추정 된다. 실습시간을 의무적으로 활용하기 위하여 실습 Lab시간 지정을 하여 해결하는 근본적인 실습체계 전환이 필요하다.

2. 프로그램 교육 개선제안

앞에서 살펴본 문제점을 고려 개선 제안의 요점은 다음과 같다.

1) 학생: 프로그래밍 학습에 뒤떨어지는 약 30%의 학생에게 필요한 것은 1:1로 도움을 줄 수 있는 프로그래밍 멘토가 매우 효과적일 수 있다. 멘토선발, 멘토교육, 멘토활용, 전체평가 시스템을 갖추고 소수 학습부진학생을 위한 멘토 시스템을 운영하는 것이 많은 도움이 될 수 있다.

2) 교재: 좋은 범용 교재 외에, 전공 특성 및 실험 실습 운영 환경에 맞는 실습교재 개발이 필요하다.

담당교수: 전임교수가 담당하고 장기적으로 맡아, 주기적 개선이 이루어 질 수 있도록 체계를 갖추어야 한다.

3) 수강학생규모: 가능한 25인 이하 소규모 인원 실습이 이루어져 맞춤형 실습이 이루어질 수 있도록 대학/학부/전공의 행재정적 지원이 필요하다.

4) 수업시간: 수업시간을 늘리는 것은 교수의 시수 및 실습실의 활용 가능성 등 근본적인 제약이 따른다. 기존체제하에서, 실습실을 체계적으로 활용할 수 있도록 조교 및 전담자를 활용한 상시 프로그래밍

