

컴퓨터 프로그래밍 교육에 적용 가능한 효과적 교수방법의 탐색적 대안

김미량[†]

요 약

효과적인 교수-학습활동의 전개는 교과교육의 방법론이 교과내용과 대상 학습자의 특성을 제대로 반영할 때 비로소 가능하다. 본 논문에서는 컴퓨터 교육내용의 핵심이 되는 프로그래밍 교육이 어떤 방법적 과정과 교수전략을 통해 진행될 때 보다 효과적일 수 있을지를 탐색해 보고자 하였다. 컴퓨터 프로그래밍 교수의 주요 구성요소에는 언어적 특징, 설계 기능, 일반적 문제 해결 능력을 들 수 있으나 특히 프로그래밍 교수설계시에는 학습자와 학습자 환경의 분석이 중요하다. 프로그래밍 교수 질의 제고를 위해서는 구조화된 강의계획서가 필수적이며, 최종 프로젝트 구성·개발방식, 문제기반 또는 해결학습법, 협동학습, 동료교수, 사례연구 등과 같은 다양한 교수방법들을 상황에 따라 융통성 있게 적용할 필요가 있다. 그 중에서도 현 시점에서는 과정적, 형성적 평가의 개념을 도입하여 수업의 전 과정을 모니터링하고 각 과정별 결과물이 누적되어 최종 프로젝트로 연결되는 방법이 프로그래밍 교수방법 개선에 가장 권장할 만한 접근방법이라고 할 수 있겠다.

Alternative Instructional Methods and Strategies for Effective Computer Programming Education

Mi-Ryang Kim[†]

ABSTRACT

For teaching-learning activities to be effective, the teaching methods need to reflect the contents of instructional materials and the characteristics of the learners. The purpose of this paper is to investigate the effective ways of teaching computer programming languages. The main components of teaching computer programming languages might be the characteristics of the language itself, the function of design, and general problem-solving capacity. But the analysis of the quality and environments of learners is much more important. To improve the quality of teaching computer programming languages, the structured syllabus needs to be provided and a variety of teaching methods such as project-based approach, problem-based approach, cooperative learning, peer tutoring and case study should be applied selectively, depending upon the instructional situations. In addition, procedural and constructive evaluation process needs to be developed to monitor each stage of learning and to give the guidelines of completing the course projects.

1. 문제의 제기

인터넷으로 대표되는 디지털 환경이 주도하는 새로운 지식정보시대를 주체적으로 살아가기 위

해서는 아날로그 환경에서와 다른 방식의 의사소통 능력이 필요하다. 국가차원에서도 경쟁력 강화를 위한 디지털 리터러시(Digital Literacy) 능력의 제고와 함께 인터넷 강국으로서의 면모를 갖추기 위해 다각적인 노력을 기울이고 있다.

이러한 노력의 일환으로 전국 대학에 컴퓨터교육과가 설치되었고, 이를 계기로 하여 컴퓨터 교

[†] 종신회원: 성균관대학교 컴퓨터교육과 교수
논문접수: 2002년 6월 6일, 심사완료: 2002년 7월 3일

육에 대한 다양한 관심과 필요성이 제기되고 있다. '세계에서 컴퓨터를 가장 잘 쓰는 국민으로 만들겠다'는 대통령의 의지가 컴퓨터교육을 위한 교육 인프라 구축에 박차를 가하면서 2002년 3월 현재까지, 교단선진화 기자재를 비롯, 하드웨어 보급 및 네트워크로 연결된 교육망의 설치 등, 컴퓨터교육을 위한 최소한의 물리적 환경이 빠른 속도로 구성되어 왔다.

그러나 이러한 여러 매체와 시설을 수업환경에 적절하게 활용하기 위한 컴퓨터교육의 제 방법이 충분히 모색되지 못하고 또 현장에서의 구체적 적용 과정에서 대상이나 내용 또는 학습자 수준 정도 등에 따라 차별화되지 못하고 있어 컴퓨터교육의 효과와 효율을 떨어뜨리는 결과를 초래하고 있다. 따라서 컴퓨터교육의 중요성과 필요성이 충분히 인식되고 있음에도 교육과정에서의 방법론 결여로 컴퓨터교육이 실효를 거두고 있지 못한은 시급히 해결되어야 할 문제로 인식된다 [3].

더욱이 이러한 현장 컴퓨터교사를 양성하는 대학 컴퓨터교육의 방법론이 새롭게 탐색·접목되지 못한다면 예비 컴퓨터 교과교사 또한 동일한 교수법을 반복함으로써 컴퓨터교과교육의 특성에 적절한 교수방법 및 전략을 개발·적용하기란 매우 어려운 문제로 남겨질 수밖에 없을 것이다. 또한 컴퓨터는 외국어의 경우와 같이 학습자간의 개인차나 수준차가 매우 크게 나타날 수 있을 뿐 아니라 개인의 노력이나 능력 고하에 따라 새로운 기회를 개척할 수 있는 도구가 될 수 있으므로 그 교육의 과정에서 학습자의 몰입과 동기유발을 위한 처방적 교수전략이 적극적으로 고안될 필요가 있다.

교과교육의 방법론이 교과내용과 대상 학습자의 특성을 반영할 때 보다 효과적인 교수-학습활동이 전개될 수 있음을 전제로, 본 연구에서는 컴퓨터 교육내용의 핵심이 되는 프로그래밍 교육에서 어떤 방법적 과정과 교수전략을 통해 교육이 진행될 때 보다 효과적일 수 있을지를 탐색·분석하여 대학 컴퓨터 프로그래밍 교육방법의 대안을 제공해 보고자 한다.

2. 효과적인 교수방법의 주요 요인

컴퓨터 프로그래밍 교육에 효과적인 방법 및 전략을 탐색·제안하기 위해서는 우선 일반적으로 '효과적인 교수활동(effective teaching)' 전개에 어떤 요소들이 반드시 필요한지를 선행연구를 통해 확인해 볼 필요가 있다.

교수방법론에 관심을 보인 선행연구들이 제안하는 효과적인 교수에 영향을 미치는 주요 요인 및 활동틀로는, i) 수업전달의 명확성, ii) 수업진행의 다양성, iii) 교수자 본연의 임무에의 충실 정도, iv) 학습과정에서의 몰입, v) 학습자 성공활동을 들 수 있는가 하면[4], vi) 역동적인 상호작용의 전개, vii) 학습자의 적극적인 참여 유도, viii) 구조화된 교수자료, ix) 학습자료 및 경험의 공유와 교환, x) 상호 협조와 네트워킹의 활성화 등 또한 양질의 교수전개에 영향을 미친다. 더불어 xi) 즉각적인 피드백, xii) 학습자의 능동적, 자기주도적 학습, xiii) 학습자에 대한 높은 기대치 설정 및 전달 등이 포함될 수 있다[9].

이상의 요소들을 통합하여 양질의 교수를 위해 교수자에게 요구되는 중요한 교수활동을 정리해 보면, 교수자는 끊임없이 보다 나은 교수전략을 고안·적용해야 하고, 모범이 되는 예시적 모델을 제공하기 위해 반성적 성찰을 실천해야 함을 알 수 있다.

3. 프로그래밍 교수에의 적용

3.1. 프로그래밍 교수의 목적

앞 절에서 간략히 살펴본 일반적인 교수법의 원리가 컴퓨터 프로그래밍 교수과정에 어떻게 적용될 수 있을까? 과연 컴퓨터 프로그래밍을 보다 효과적으로 가르치는데 가장 적합한 방법론이 존재할까? 존재한다면 어떤 방법이 가장 적절할까? 이러한 연속적 의문의 실마리는 프로그래밍 교수의 결과, 학습자가 획득할 수 있는 능력에는 어떤 것들이 있을지를 파악해보는 것으로부터 출발

할 수 있다. 이는 곧 프로그래밍 교수의 궁극적 목적이 되기도 할 것이다.

예컨대, 프로그래밍 교수의 과정을 통해 학습자가 분석력, 논리적, 적용력, 응용력, 창의력, 이해력, 통찰력, 사고력, 예측력, 조직력, 판단력, 문제해결력 등을 갖추고 성장시켜 나갈 수 있기를 기대할 수 있다. 그러나 이러한 고차적 능력들이 프로그래밍 교수를 통해 성공적으로 함양되기 위해서는 이들 능력을 개발하기 위한 프로그래밍 교수 교유의 방법과 전략이 고안, 통합되어야 한다.

3.2. 프로그래밍 교수의 주요 구성요소

이상과 같은 고차적 사고기능을 획득할 수 있는 컴퓨터 프로그래밍 교수의 주요 구성요소는 i) 언어적 특징, ii) 설계 기능, iii) 일반적 문제 해결 능력으로 요약될 수 있다[10].

즉 일차적으로는 해당 언어의 특징을 파악하여 필요한 내용을 숙지할 필요가 있고 각각의 기능이 어떻게 실행되는지 설계 및 실행과정을 통해 테스트해 보며 이러한 과정을 통해 단위 모듈별로 일련의 문제를 성공적으로 해결할 수 있도록 반복적으로 연습하는 과정을 거치게 된다.

그렇다면 이러한 프로그래밍 교수의 특징적 요소가 어떻게 수업장면에 통합될 수 있을까? 그 구체적 과정에 대한 창조적 고민이 필요하다.

3.3. 프로그래밍 교수설계 연습

이전에도 다수의 교수자들이 강의계획서를 통해 한 학기의 수업을 사전에 계획하고 그 일정에 따라 수업을 전개해 왔으나, 최근에는 대부분 대학의 행정이 종합정보시스템과 같은 학사일정 관리 그룹웨어를 통해 전자화되면서 학습자의 수강신청을 위한 강의계획서의 사전 입력이 필수화되고 있는 추세이다. 엄밀한 의미에서 강의계획서 작성은 처방적 교수설계의 출발점이라고 할 수 있을 정도로 중요한 의미를 갖는다.

따라서 Dick과 Carey(2001)의 체계적 접근(systematic approach)에 따라[5], 각 단계별로 프로그래밍 교수과정을 설계해 보면 보다 체계적이

고 절차적인 프로그래밍 수업을 전개하는 데 유용한 도움을 얻을 수 있다.

예를 들어 한 학기동안 자바언어의 기초를 가르친다고 가정하고 그 과정을 Dick & Carey의 단계에 따라 다음 [그림 1]을 참고하여 설계해 보도록 하자.

[그림 1] Dick & Carey의 단계에 따른 설계연습

각 단계별로 자바(해당 프로그래밍 언어) 교수에 적절한 내용을 설계·기록해 나가면 되나 특히 프로그래밍 교수에서 가장 중요하게 고려되어야 할 측면은 바로 학습자 분석의 단계라고 할 수 있다.

학습자 분석에서는 자바 프로그래밍 수업에 참여하는 개별 학습자의 출발점 행동 및 준비도를 파악하는 것이 반드시 필요하다[5]. 특히 프로그래밍 언어 교수에서 가장 빈번하게 접하게 되는 어려움으로는 학습자간의 현격한 개인차이므로, 필요한 경우, 이들의 사전능력의 정도를 진단 평가해 보는 것도 추천할 만한 방법이라고 할 수 있다. 이 과정에서 확인하게 되는 개별 학습자의 다양한 수준차를 고려하여 최종 프로젝트 팀을 구성하게 되면 학습자간의 상호 동료학습(peer tutoring)의 경험과 폭을 심층·확대할 수 있게 된다. 또한 학습자 분석과정에서 프로그래밍 관련 사전 학습의 경험이나 히스토리(history)를 확인하고 컴퓨터 관련 기능의 숙달 정도를 점검하는 것도 유용한 도움이 될 수 있다.

이러한 학습자 분석의 중요성은 최근 교수-학습활동에 구성주의적 관점이 강력하게 부각되면서 학습자 중심의 학습환경 구성에 대한 관심이 점차 확대되고, 또 개인 학습자도 자신의 필요와

눈높이에 따라 자기주도적(self-directed), 자기규제적(self-regulated) 학습을 전개하려는 학습의지가 강화되고 있는 추세와 맥을 같이 한다고 볼 수 있다.

학습자 중심 환경의 구성은 웹을 통한 사이버 교육의 방법이 활성화되면서 더욱 중요한 이슈가 되고 있는데, 이러한 사이버 공간은 면대면 교수-학습환경의 보조적 수단으로도 그 활용의 간소성이 점차 확대되고 있다. 특히 프로그래밍 교수와 같이 컴퓨터(언어)를 가르치는데 컴퓨터 기반 환경을 활용하는 것은 양자간의 상승적 보완작용을 할 수 있다는 측면에서도 장려될 만한 교수방법이라고 할 수 있다.

[그림 1]의 Dick & Carey의 접근이 교수자의 입장에서 사전에 교수를 설계해 나가는 전통적 과정을 연습해 보는 것이라면 이러한 과정을 통해 준비된 각 단계의 활동을 실제 수업의 장면에서 전개할 때에는 보다 구체적인 시나리오가 필요하게 된다. 예컨대, [그림 2]에 제시된 Gagne의 9가지 교수사태(instructional events)가 유용한 지침을 제공해 줄 수 있다. Gagne의 외적 사태는 효과적인 교수를 전개하는 교수자의 일련의 교수 활동을 과정적으로 구조화해 놓은 것으로 대부분의 교수활동이 이러한 사태를 기초로 진행되나, 프로그래밍 교수의 각 사태별로 필요한 활동들을 추출하고 이 과정을 연습해 봄으로써 교수 구성에 보다 구체적인 안내를 제공해 줄 수 있을 것으로 기대된다. [그림 1]의 연장선 상에서 동일한 가정(자바언어기초강좌)으로 [그림 2]의 공란을 완성해 볼 수 있을 것이다.

이 과정은 비교적 교수설계의 기본적인고도 전 통적인 절차를 예시한 것에 불과하나 이러한 연습 과정을 통해 교수활동 구성의 각 요소나 단계별 활동을 숙고·고안해 봄으로써 보다 효과적인 프로그래밍 교수활동을 구체화할 수 있을 것으로 기대된다.

4. Syllabus 구성 및 개발 연습

4.1. 구조화된 프로그래밍 Syllabus의 구성 요소 탐색

앞 절의 과정을 통해 효과적인 프로그래밍 교수설계의 기본 절차를 연습해 보았다면 이제는 완성도 높은 체계적 Syllabus를 구성·개발해 보는 과정을 통해 프로그래밍 교수의 질을 제고하는 방안을 살펴보기로 하자.

전술한 바와 같이 대학행정의 종합정보시스템 도입으로 강의계획서의 사전 입력이 이미 보편화 되고 있으나, 초중고 교육현장에서 널리 활용되었던 수업지도안이 실제 수업을 위한 지도안이라기보다는 행정 제출용 지도안으로 작성된 경향이 있었던 과거의 경험에 비추어 대학의 강의계획서 또한 입력용 계획서에 불과할 가능성 또한 배제할 수 없을 것으로 예상된다.

따라서 실제 교수과정에 활용가능한 Syllabus를 구성·개발해 보는 것은 효과적인 교수의 중요한 전제라고 할 수 있다.

그렇다면 잘 구조화된 Syllabus는 어떤 요소로 구성될 수 있을 것인가? 교수자라면 누구나 '잘 가르치고' 싶은 의욕이 있을 것이고 교수평가를 통한 피드백이 긍정적이고 만족스럽기를 기대할 것이다.

이 때 자신의 교수의 질을 개선하는데 중요한 자극이 되는 것으로 학습자의 진솔한 피드백과 'Best Practice'의 예를 참조하는 것을 생각해 볼 수 있다. 한 마디로 '명강의'를 칭강해 볼 수 있는 기회가 있다면 그 교수자의 행동을 구조화하거나 모사해 보는 것도 교수방법 개선에 실제적

도움이 될 수 있다는 의미이다. 단적으로 초중고 교육현장에서는 '연구수업'의 형태로 교사들의 수업이 공개되고 평가받는 기회가 있어 이에 임하는 해당 교사는 매우 진지하게 수업을 준비하게 된다. 이 때 학생들은 물론 수업을 참관한 동료 교사도 건설적 피드백을 제공할 수 있기 때문에 수업의 질 개선에 도움이 되며, 또한 효과적인 교수방법이나 오랜 경험에서 획득된 암묵적 노하우가 표출, 발현되는 사례를 발견, 공유할 수 있는 기회가 되기도 한다. 즉 나와 동일한 교과내용을 가르치는 다른 교수자는 어떻게 수업을 구성하고 전개해 나가는가에 대한 궁금증이 누구나에게 있을 수 있다는 것이다.

프로그래밍 교수의 Syllabus 구성은 이러한 본연적 의문을 해결하는 것에서 출발해 볼 수 있다. 다시 말해서 효과적인 교수를 전개하는 다른 교수자의, 잘 구조화된 Syllabus를 벤치마킹하여 그 과정을 면밀히 분석해 보는 것도 효과적인 교수를 위한 하나의 방법이라고 할 수 있겠다.

그렇다면 프로그래밍 언어 강좌의, 잘 구조화된 Syllabus는 어떤 요소로 구성되어 있을까? 왜 이 Syllabus가 잘 구조화되었다고 생각되는가? '나'의 Syllabus와는 어떤 차이가 있는가? 등의 의문을 가지고 <표 1>에 수록되어 있는 다른 프로그래밍 강좌의 Syllabus를 한 번 살펴해보도록 하자.

<표 1> 프로그래밍 관련 강좌의 Syllabus사이트

- <http://www.math.grin.edu/~rebelsky/Courses/Tutorial/99F/Handouts/syllabus.html>
- http://kcweb.nhmccd.edu/programs/tech/csci/distance_Internet/_CPP/csci1433_29051.htm
- <http://sos.heinz.cmu.edu/courses/java/syllabus.html>
- <http://carbon.cudenver.edu/public/wle/phpcourse/syllabus.php>
- <http://www.cs.arizona.edu/people/greg/mpdbook/lectures/>
- <http://www.cs.arizona.edu/classes/cs227/spr01/227Syllabus.html>

이상의 예시적 사이트를 비롯, 여타 프로그래밍 강좌의 Syllabus를 참고로 하여 잘 구조화된

Syllabus의 대표적인 구성요소를 추출해 보면 다음과 같은 요소를 확인해 볼 수 있다[11] [12][13] [14].

- i) 강좌의 소개/개요 및 개설의 목적,
- ii) 소정의 교수목표 및 학습결과 안내
- iii) 교수자 정보 및 의사소통 채널
- iv) 주 교재 및 보충 교재 목록
- v) 프로그래밍 실습을 위한 H/W, S/W 환경
- vi) 대상 학습자
- vii) 강좌 안내: 과제, 평가, 면담시간, 시험일정
- viii) 학습참여의 방법 및 혜택
- ix) 강의노트(웹사이트와 연동)
- x) 상세한 학습일정 및 예상 진도
- xi) 관련 사이트
- xii) 평가시의 유의사항

경우에 따라서는 수강취소의 방법이나 과제 제출 마감을 지키지 못했을 때의 감점 기준 등의 정보까지도 상세히 안내하고 있는 Syllabus를 발견할 수 있었다[15].

교수자에 따라 어떻게 구조화하는가의 차이가 있을 수 있겠으나 기본적으로 Syllabus에 포함되어야 할 기본적인 요소들은 교수-학습의 구성요소를 포괄해야 하며 기타 학습자의 학습을 촉진, 안내하는 관련 정보 및 웹 사이트의 제공도 권장할 만한 사항이다.

4.2. '나'의 Syllabus 재설계와 구조화

<표 1>의 사이트는 프로그래밍 관련 강좌를 검색 엔진을 통해 탐색하여 대표적인 예를 제시한 것으로, 가능하면 다양한 프로그래밍 언어의 Syllabus를 예시하고자 한 의도일 뿐이며 이 사이트에 포함된 Syllabus가 잘 구성·설계되었음을 의미하지는 않는다. 다만 제공자별로 Syllabus 작성의 방법이나 구성요소, 안내의 정도 등이 매우 다양하게 나타나고 있음을 확인할 수는 있을 것이다. 이들 예를 바탕으로 Syllabus에 반드시 포함되어야 할 구성요소들을 확인·추출하고 자신의 수업을 위한 Syllabus를 보다 체계적으로

구조화하여 재작성해 보는 것도 보다 효과적인 교수에 도움이 될 수 있을 것이다. 이 때 수업진행의 일정을 가능한한 상세화하고 내용에 따라 융통성있게 적용될 수 있는 최적의 교수방법과 전달 매체 등에 대한 정보도 함께 제시함으로써 교수 전개에 방향을 보다 친절하게 안내할 수 있다.

5. 효과적인 프로그래밍 교수방법의 탐색

5.1. 학습자가 보고하는 현 프로그래밍 교수 전개의 문제점

경험적인 데이터로 근거 자료를 제시하기는 어려우나 프로그래밍 강좌를 수강한 다수의 학습자들의 일성은 대부분 처음 의도했던, 즉 목표로 설정했던 분량의 학습이 진행되지 못했다는 것이다. 그래서 여러 프로그래밍 언어를 학습할 기회를 갖기는 하되, 끝까지 한 언어라도 숙지하는 경우가 드물고 주요 기능 중심으로 수업이 진행되기 때문에 완성도 있는 프로그램을 구성하기 어렵다는 것이다. 그 만큼 하나의 프로그래밍 언어를 숙지하는 데에는 많은 시간과 노력이 요구된다는 의미일 것이나 프로그래밍 교수방법의 개선이 촉구되는 대목이기도 하다.

또한 학습자의 수준차와 관계없이 동일한 과제가 부과되는가 하면, 소수 학습자 중심으로 수업이 진행되어 중반이후에는 많은 학습자들의 수업 참여도가 현저히 떨어지는 경우가 많다고 보고하고 있다. 이와 같은 중도탈락은 학습자의 개별 수준을 무시하고 획일적, 일방적으로 전달되는 수업내용이 다양화되지 못한 결과로 비롯된 것으로, 이러한 결과는 극단적으로 사설 학원기관을 통해 프로그래밍 기능의 향상을 시도하는 학습자를 양산하고 있다.

나아가 현행 프로그래밍 교수방법의 가장 결정적인 문제로 지적될 수 있는 것은 해당 프로그래밍 언어를 활용한 최종 결과물이 개별적으로 또는 소집단별로 완성된 경우가 매우 드물다는 사실이다. 각 단계별, 모듈별로는 해당 언어를 숙지

하는 학습이 잘 전개되나 거시적, 통합적 접근이 부족하여, 해당 언어를 배우고 나서 그 언어로 학습자가 무엇을 할 수 있게 되었는가의 측면에서는 뚜렷한 성과를 제시하기 어렵다는 점이다. 교수-학습활동의 효과를 결과적 측면에서만 측정, 판단할 수는 없다고 하더라도 프로그래밍 교수의 특성상 최종 교수활동이 완료된 이후에 학습자가 획득하게 된 프로그래밍 능력의 정도와 완성도, 산출물 등은 교수-학습과정의 질을 판단하는 중요한 척도가 된다고 할 수 있으므로 이상에서 지적된 현행 프로그래밍 교수방법의 문제들은 시급히 보완되어야 할 측면이라고 할 수 있겠다.

5.2. 프로그래밍 교수에 적용가능한 대표적 교수방법

컴퓨터 프로그래밍 교수에 주로 활용되고 있는 대표적인 교수방법으로는 i) 강의법/설명법, ii) 시연(Demonstration), iii) 실습(Drill & Practice), iv) 질문법과 토론법 등을 들 수 있겠다[1][3].

강의법 및 설명법은 교수-학습활동에 가장 기본이 되는 방법으로서[2], 대학의 교수활동이 '강의', '강의실'로 대표되는 것으로 미루어 볼 때 대학교육에서 가장 빈번하게 활용되는 교수방법으로 간주될 수 있을 것이다. 그러나 '강의'가 여러 다양한 교수방법의 한 방법임을 감안하고 또 대학의 교수활동이 주로 '강의'로 지칭되면서 오히려 교수방법을 '강의'로만 국한하고 있는 것은 아닌가를 의심해 보면서 대학의 교수방법을 보다 다양화할 필요를 촉구하게 된다.

시연은 현재 빔 프로젝터와 같은 고가의 투사장비가 대학교육환경에 급속히 보급되면서 많이 적용되고 있는 교수방법으로[2][3], 특히 컴퓨터교육에서는 교수자의 시연과정 후, 학습자의 개별 실습이 진행되는 방식으로 빈번하게 활용되고 있는 방법이라고 할 수 있다.

특히 실습은 컴퓨터교육에서 반드시 적용되어야 하는 교수-학습방법으로서[1][3], 이를 위한 컴퓨터 실습실이 반드시 구비되어 있어야 하는데 컴퓨터의 성능과 관리상태, 사용시간에 따라 교

수-학습의 효율을 좌우하므로 물리적 환경에 대한 사전 점검이 반드시 필요하다.

그 외에 질문법과 토론법은 일반적으로 활용되는 교수방법이나[3][8], 이러한 방법들이 프로그래밍 교수환경에서 얼마나 유용하고도 적절하게 활용되어 학습자의 학습을 촉진할 수 있을지에 대해서는 보다 심층적인 고민과 적용이 필요하다. 프로그래밍 교수의 결과, 궁극적으로 획득될 수 있는 여러 능력들을 상기해 볼 때 문제해결력이나 창의력 등 고차적 사고기능을 지향하고 있음을 고려하여 학습자 스스로 프로그래밍 언어 기능의 숙지과정에서 질문의 생성과 해결, 동료 학습자와의 토론과 의견교환 등이 활성화되어야 할 것이다.

5.3. 효과적인 프로그래밍 교수방법의 예

앞 절에서 살펴본 교수방법들이 대체로 전통적, 일반적 방법들이라고 할 수 있다면 우리는 효과적, 효율적, 매력적인 프로그래밍 교수를 위해 보다 적합한 교수방법을 탐색하고 또 탐색한 방법 또한 대상 학습자 집단에 적절하게 맞춤형 하는 노력을 기울일 필요가 있다.

예를 들어 프로그래밍 교수활동에 보다 효과적일 수 있는 교수방법으로는, i) 최종 프로젝트 구성·개발(Project-Based Approach), ii) 문제기반 또는 해결학습(Problem-Based/Problem-Solving Learning), iii) 협동학습(Cooperative Learning), iv) 동료교수(Peer Tutoring), v) 사례연구(Case Study) 등이 제안될 수 있겠다.

프로그래밍 교수에서 Project-Based Approach는 가장 적극적으로 확대, 적용되어야 하는 교수방법으로서, 최종 결과물이 반드시 완성되어야 하되, 그 결과물을 산출해 가는 과정적 단계에서의 피드백과 형성평가의 과정이 모니터링되어야 한다. 하나의 프로젝트를 개발해 가는 과정에서 학습자는 비록 습작일 수는 있겠으나 해당 언어로 하나의 완성물을 구현해 가는 만족과 성취감을 동시에 만끽할 수 있게 된다. 즉, 수업 초기 단계에서부터 최종적으로 설계, 개발될 프로젝트 산출물의 주제를 선정하고 수업이 진행되면서 각 단계별로 해당 모듈을 완성해 가며 최종 산물로

통합해 가게 된다. 이 과정에서 학습자간에 적극적인 피드백을 통해 최종 결과물의 질을 개선해 나갈 수 있다. 더불어, 이러한 프로젝트가 누적될 때 포트폴리오의 형태로 관련 자료를 수집하여 개인 학습자의 컴퓨터 관련 능력의 성장사를 기록하고 모니터링하는 것도 추천할 만한 관리 방법이라고 할 수 있다.

문제해결 학습의 방법은 프로그래밍 교수가 추구하는 궁극적인 결과 중, 문제해결능력을 신장시켜 나가는 것이 중요한 비중을 차지하고 있으므로 이 또한 간과할 수 없는 중요한 교수-학습 방법이라고 할 수 있다. 즉 우선,

- i) 문제를 규명하고,
- ii) 문제를 연구, 분석하여 문제의 본질을 도출한 다음,
- iii) 다양한 가능성을 검토, 비교하여 취사선택하며,
- iv) 관련 노력을 실천하고,
- v) 이전의 입장과 위치를 수정하기 위해 학습의 결과를 반추하는 과정을 거치게 된다.

이상과 같은 일련의 문제해결 과정을 프로그래밍 교수의 과정에 유연하게 적용하게 되면 학습자는 자기주도적으로 프로그래밍을 통한 문제해결 과정을 반복, 연습하게 됨으로써 단순한 프로그래밍 상의 에러나 버그를 발견, 수정하는 것 이상의 고차적 사고훈련도 동시에 수행할 수 있을 것으로 기대된다. 따라서 문제해결 학습은 프로그래밍 교수방법으로 적극 권장할 만한 교수방법이라고 할 수 있다.

협동학습은 이상의 방법들과 통합적으로 적용 가능한 방법으로서[7], 특히 개인 학습자간의 컴퓨터 활용능력의 차이가 현저할 때 상호 협조적으로 적용할 수 있는 방법이라고 할 수 있다. 다만 이 때의 협동학습은 학습의 개별화를 전제로 개인 학습자가 자신이 협동학습에 참여하는 다른 학습자에게 공헌할 수 있는 측면을 끊임없이 강구해야 한다.

동료교수(peer tutoring)나 사례연구는 일대대의 실습실 또는 강의실 교수환경에서 발생할 수 있는 학습결손을 방지하고 동일한 입장과 눈높이에서 서로에게 유용한 도움과 정보를 제공하는

방법으로[6], 구성주의에 기초한 학습자 중심의 학습환경 구성에 대한 관심이 부각된 이후, 현재 매우 활성화되고 있는 교수방법이라고 할 수 있다. 프로그래밍 교수에 적극적으로 활용될 수 있는 유용한 방법으로 추천되며 이러한 방법들이 사이버 공간에서 적용될 수 있을 때 그 영향력과 효율성이 더욱 배가되어 여러 학습자가 동시에 도움을 받을 수 있는 학습채널이 형성될 수 있다.

5.4. 시연형 교수자에서 촉진형 교수자로

'시연'은 현재 프로그래밍 교수자의 대다수가 채택하고 있는 대표적인 교수방법 중의 하나라고 할 수 있을 것이다. 시연형 교수자는 대개 설명, 시연, 참여, 교차, 시험, 종결, 전이 등의 과정을 따라 교수활동을 전개한다.

그러나 학습자 중심의 프로그래밍 교수활동을 전개하기 위해 교수자는 당연히 '촉진형 교수자'로 거듭날 필요가 있다. 촉진형 교수자는 구체적인 예를 활용하여 학습자로 하여금 반추적 관찰을 할 수 있도록 안내하고 학습자 스스로 개발한 추상적 원리를 기초로 실제적인 아이디어를 접목, 실천하도록 촉진하는 역할을 하게 된다.

프로그래밍 교수과정에서도 교수자가 먼저 나서서 무엇인가를 끊임없이 가르쳐야 한다고 생각해 왔던 강박관념에서 벗어나, 이제는 학습자의 내재적인 자기주도적 학습능력과 창조적 발상을 기대하고 학습자 스스로의 적극적인 사고활동을 촉진하는데 보다 심혈을 기울일 시점이라고 할 수 있겠다.

6. 적용가능한 방법의 통합적 접근

지금까지 컴퓨터 프로그래밍 교육에 적용 가능한 여러 방법들을 탐색, 제안해 보았으나 그 어떤 방법도 한 두가지의 방법만으로는 효과적이고 효율적이며 교수자·학습자 양자 모두가 만족하는 수업을 전개하기는 매우 어려울 것이라고 단언할 수 있다.

특히 교수방법은 교육의 내용과도 밀접한 관련

이 있지만 대상 학습자에 따라 매우 융통성있게 적용·접목되어야 하는 특성을 지니므로 보편적, 일반적 상황에 적절한 처방적 대안을 제시하는 것은 매우 어려운 일이라고 할 수 있겠다.

따라서 효과적인 프로그래밍 교수법을 위해서는 교수활동 전개의 각 과정에서 유용한 방법론이 상황에 따라 융통성있게 적용되어야 하되, 과정적, 형성적 평가의 개념을 도입하여 수업의 전 과정을 모니터링하고 각 과정별 결과물이 누적되어 최종 프로젝트로 연결되는 방법이 현 시점에서 프로그래밍 교수방법 개선에 가장 권장할 만한 접근방법이라고 할 수 있다. 특히 최종 프로젝트 개발과정에서의 산출물은 물론 결과물의 내용을 수업에 참여한 학습자 모두가 공유함으로써 상호 피드백을 통한 건설적 평가활동이 활성화될 때 동료교수(Peer Tutoring)의 효과도 동시에 획득될 수 있을 것이다. 다만 이 과정에서 활동과제나 퀴즈 등을 적극 활용하여 학습자의 진행상황을 수시로 진단할 필요가 있다.

또한 교수자와의 의사소통 채널이 확보되어야 하며 항상 제기된 질문에 대한 신속하고 명확한 답변이 제시되어야 한다. 이러한 교수-학습의 양방향 의사소통이 원활하게 진행되기 위해서는 사이버 상에 수업을 위한 공간을 개설하여 면대면 교수-학습상황에서 채 해결되지 못한 여러 문제들을 공유하고 해결방안을 강구하는 노력을 축적해 나가는 것도 디지털 환경에서의 교수활동에 적용되어야 할 측면이라고 할 수 있겠다.

참 고 문 헌

- [1] 이무근, 김재식, 김판욱(2000). 실기교육방법론. 서울: 교육과학사.
- [2] 이성호(1999). 교수방법론. 서울: 학지사.
- [3] 허희옥의(2001). 컴퓨터교육방법 탐구. 서울: 교육과학사.
- [4] Borich, G.D.(2000). Effective teaching methods(4th Ed.). NJ:Merrill/Prentice Hall.
- [5] Dick, W., Carey, L., & Carey, J.O.(2001). The systematic design of instruction(5th

Ed.). NY: Addison-Wesley Educational Publishers Inc.

- [6] Goodlad, S., & Hirst, B.(1989). Peer tutoring: A guide to learning by teaching. NY: Nichols Publishing.
- [7] Johnson, D.W., & Johnson, R.T.(2000). Cooperative learning. available at <http://www.clcrc.com/pages/cl.html>
- [8] Morgan, N., & Saxton, J.(1991). Teaching, questioning, and learning. NY: Routledge.
- [9] <http://ag.arizona.edu/azlearners/aahe-7principles.html>
- [10] <http://www.holtsoft.com/turing/essay.html>
- [11] <http://www.math.grin.edu/~rebelsky/Courses/Tutorial/99F/Handouts/syllabus.html>
- [12] http://kcweb.nhmccd.edu/programs/tech/csci/distance_Internet/PHP/csci1433_29051.htm
- [13] <http://sos.heinz.cmu.edu/courses/java/syllabus.html>
- [14] <http://carbon.cudenver.edu/public/wle/phpcourse/syllabus.php>
- [15] <http://www.cs.arizona.edu/classes/cs227/spr01/227Syllabus.html>
- [16] <http://www.cs.arizona.edu/people/greg/mpdbook/lectures/>
- [17] <http://www.cs.ukc.ac.uk/pubs/1997/208/index.html>

김 미 랑

1987 서울대학교 인문대학

영어영문학과(문학사)

1989 미국 리하이대학교 대학원

교육공학과(이학석사)

1998 서울대학교 대학원 교육학과

교육방법 및 교육공학 전공(교육학박사)

1998~1999 서울대학교 교육연구소 특별연구원

1999~현재 성균관대학교 사범대학

컴퓨터교육과 교수

관심분야: 컴퓨터 기반의 교수-학습환경 설계 및 개발, 컴퓨터교육, 컴퓨터 통신·인터넷을 활용한 사이버교육, 혁신의 확산

E-Mail: mrkim@comedu.skku.ac.kr