

웹 기반 프로그래밍 과목의 효과적인 강의 전략

이충기[†] · 홍석원^{†*}

요 약

언제 어디서나 적은 비용으로 프로그래밍을 배우고자 하는 수요가 늘어남에 따라 웹 기반의 프로그래밍 강의가 많이 이루어지고 있다. 일반적으로 웹 기반으로 프로그래밍을 가르치는 것은 매우 어렵다. 따라서 이 강의를 효과적으로 가르치기 위한 방법들에 대한 연구가 필요하다. 본 논문은 관련 문헌들과 수년 간에 걸쳐 웹 기반으로 프로그래밍 과목을 가르친 경험에 기초하여 효과적인 웹 기반의 프로그래밍 강의 전략을 제시한다. 먼저 프로그래밍 교과목의 특성과 학습자들의 특징을 고려하고 프로그래밍 능력을 개발하기 위한 교안 설계 전략을 제시한다. 다음으로 그 과목의 효과적인 평가 및 운영 전략을 제시한다. 마지막으로 제안된 전략들에 따라 진행된 최근 강의의 성적 평가와 운영 결과를 분석한다. 이는 웹 기반 강의의 교육 효과를 높일 수 있음을 보여 준다.

키워드: 웹 기반 강의, 프로그래밍

Effective Strategies for Teaching a Web-based Programming Course

Chungki Lee[†] · Sukwon Hong^{†*}

ABSTRACT

The demand for learning programming anywhere at a low cost at any time is increasing. Accordingly, there have been a great number of web-based programming courses. Generally, teaching Web-based programming classes is very challenging. Thus effective methods for teaching these courses should be researched. Based on several years of experiences in teaching a Web-based programming course as well as the related literature, this paper presents its effective teaching strategies. First, some design strategies of its course contents for developing programming skills are proposed, considering the characteristics of programming courses as well as potential learners. Next, its effective assessment and class management strategies are proposed. Finally, we evaluate the results of student grades and operation of a Web-based programming course offered most recently using proposed strategies. They demonstrate that we can enhance learning effects of the course.

Keywords : Web-based Teaching, Programming

1. 서 론

방송, 컴퓨터와 통신망을 이용한 원격 교육은 지난 수십 년 동안 이루어져 왔다. 원격 교육은 일부 성과를 거두기도 하였으나 교육 효과가 전 통적인 면 대 면 강의에 비해 낮았기 때문에 보

† 정회원: 명지대학교 컴퓨터소프트웨어학과 교수(교신저자)
 † 정회원: 명지대학교 컴퓨터소프트웨어학과 교수
 논문접수: 2004년 2월 26일, 심사완료: 2004년 4월 16일

편화되지 못하고 제한된 여건과 환경에서만 활용되었다. 최근에 인터넷과 웹의 이용이 보편화됨에 따라 이를 교육에도 활용하고자 하는 많은 노력이 전 세계적으로 교육자들에 의해 이루어지고 있다. 그 결과로 인터넷과 웹을 이용한 교육이 주요 연구 분야중의 하나가 되었다[7][9]. 현재 웹 기반 교육이 새로운 강의 방법으로 자리 잡아가고 있다. 대부분의 대학들이 일부 강의를 인터넷으로 제공하고 있으며 모든 강의를 인터넷으로 진행하는 국내의 사이버 대학들이 개교하여 성공적으로 운영되고 있는 것이 좋은 예이다. 기업에서도 가상 교육을 통한 재교육이 이미 실시되고 있다[8]. 빠른 기술 발전에 따라 직장인들의 재교육 필요성이 늘어나고 언제 어디서나 적은 비용으로 교육을 받고자 하는 수요가 증대되고 있으므로 웹 기반 강의의 전망은 매우 밝고 많은 잠재 고객을 가지고 있다.

컴퓨터의 이용이 보편화되고 거의 모든 정보처리가 정보시스템화 됨에 따라 이 시스템을 개발하는 프로그래머의 수요가 증가하고 있다. 또한 정보기술의 빠른 발전은 철저한 프로그래밍 교육에 대한 필요성을 더욱 높이고 있다. 그에 따라 언제 어디서나 적은 비용으로 프로그래밍을 배우고자 하는 수요가 늘어나고 있다. 이러한 수요에 따라 웹 기반의 프로그래밍 강의를 국내외 사이버대학과 정보기술 전문 교육기관 등에 의해 이루어지고 있다. 초보자들에게 쉽게 프로그래밍을 가르치기 위한 많은 학습(tutoring) 시스템들이나 웹 환경 시스템들이 개발되었다[6,11,14,16]. 또한 실제 사이버 프로그래밍 교육/학습 사례도 많이 발표되고 있다[12,13,15]. 그러나 일반적으로 프로그래밍을 가르치는 것은 매우 어렵다. 프로그래밍 언어가 우리에게 익숙하지 않은 인공적인 언어이고 프로그래밍을 하기 위해서는 많은 연습과 경험으로 얻어지는 지식들을 필요로 하기 때문이다. 웹 기반 강의에서는 개별적인 프로그래밍 지도가 불가능해 더 어렵다. 따라서 웹 기반으로 프로그래밍을 가르치기 위한 효과적인 방법에 대한 연구가 필요하다.

본 논문은 관련 문헌들과 수년 간에 걸쳐 웹 기반으로 자바 프로그래밍 과목을 가르친 경험에 기초하여 효과적인 웹 기반의 프로그래밍 강의

전략을 제시한다. 특히 초보 프로그래머들이 웹 기반 강의를 통해 프로그래밍을 배우는 것을 효과적으로 지원하고 도와주기 위한 강의 전략에 초점을 맞춘다. 그동안 강사들과 조교들의 경험 부족, 수강생들의 동기와 노력 부족, 수강생들의 증가와 강의 플랫폼의 제약으로 말미암아 많은 어려움과 시행착오를 겪었다. 특히 중도 탈락하거나 낙제한 학생들의 수가 많아 이를 개선하기 위한 많은 노력이 있어 왔다. 그 결과로 강의는 강의 범위의 조정, 운영 방법의 변경, 평가 방법의 변경 등 많은 변화를 겪었다. 제시된 강의 전략에 따라 운영된 최근의 강의 결과는 웹 기반 강의의 교육 효과를 높일 수 있음을 보여 준다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련된 배경 지식을 요약한다. 3장에서는 연구 방법을 기술한다. 4장에서는 프로그래밍 능력을 개발하기 위한 교안 설계 전략을 제시한다. 5장에서는 웹 기반 프로그래밍 과목의 효과적이고 실제적인 운영 및 평가 전략을 제시한다. 6장에서는 제안된 전략에 따라 진행된 최근 강의의 운영 사례를 보고하고 결과를 분석한다. 7장은 연구 결과를 요약한다.

2. 연구 배경

2.1 프로그래밍 학습 이론

프로그래밍을 배우는 것과 가르치는 것이 쉽지 않아 지금까지 많은 연구가 이루어졌고 그 결과로 프로그래밍 학습 이론들이 많이 발표되었다 [1,3,4,11,17]. de Boulay는 프로그래머가 습득해야 하는 다음과 같은 다섯 가지 영역들을 제시하고 있다[2]:

- (1) 프로그램에 대한 정확한 이해
 - (2) 프로그램의 실행과 관련된 컴퓨터의 모델
 - (3) 프로그래밍 언어의 구문론과 의미론
 - (4) 문들을 결합하여 구조화하는 방법
 - (5) 완성 프로그램을 작성하는 실제적인 방법
- 위의 영역들은 서로 중복되고 프로그램을 배우기가 어려운 잠재적인 원인이 될 수 있다. 따라서 프로그래밍 관련 지식을 배우고 개발하는 것

은 매우 복잡한 과정이다. 특히 초보자가 간단한 프로그램을 작성하기 위해서 위의 영역들을 다 알아야 하므로 많은 어려움을 겪을 수 있다.

Deimel과 Moffat는 프로그래밍 입문 과목에는 다음과 같은 네 단계가 있어야 한다고 제안하였다[1]:

(단계 1) 학생들이 프로그램의 동작을 관찰하고 그것의 이로움과 한계를 평가하기 위해 프로그램을 실행시키게 한다.

(단계 2) 학생들이 좋은 구조를 가진 프로그램들을 공부하게 한다.

(단계 3) 학생들이 모델 프로그램을 수정하고 확장하게 한다.

(단계 4) 학생들이 새 프로그램을 독립적으로 설계하고 작성하게 한다.

초보자들이 위의 첫 세 단계들을 수행하기 위해 잘 설계된 예제 프로그램들을 사용하는 것은 좋은 전략이다. 즉, 학생들은 예제 프로그램을 실행하고 수정하고 확장한다. 특히 학생들은 자주 좋은 예제 프로그램을 새 프로그램을 작성하기 위한 기초로 사용한다. 이 과정을 통해 학생들은 프로그래밍 개념들과 기술들을 배우게 된다. 이 전략은 완성 전략(completion strategy)이라고 불린다[3]. 네 번째 단계를 따르는 전략은 생성 전략(generation strategy)이라고 불린다. 이 전략은 초보자들에게 바로 적용하기가 어려워 배우고자 하는 의욕을 더욱 떨어트리거나 좌절하게 할 수 있다. 따라서 완성 전략이 초보자를 위한 더 적절한 학습 전략이라고 알려져 있다[3, 11].

Oliver는 그의 실험에 기초하여 프로그래밍 지식에 관한 계층적 구조를 제안하였다[4]. 그에 의하면 프로그래밍 지식의 다음과 같은 세 가지 유형이 있다:

- (1) 구문론 및 프로그램 구조의 이해
- (2) 문들(statements)을 적용하는 능력
- (3) 문제 해결 능력

첫 번째 유형의 지식은 연산들(operations)의 동작을 이해하는 수준을 넘어 알고리즘을 이해하고 프로그램 내에 있는 문들을 이해할 수 있는 능력을 의미한다. 두 번째 유형의 지식은 문들을 결합하여 알고리즘을 구현하는 능력을 의미한다. 이 지식을 가진 학생들은 모델 프로그램과 비슷

한 해법을 가지는 문제에 대해 주어진 프로그램을 이용하여 풀 수 있어야 한다. 세 번째 유형의 지식은 초보자의 제한된 지식으로 인해 갖추기가 어렵다. 따라서 그는 초보자들이 프로그램을 독립적으로 작성하는 것을 요구하기보다 모델 프로그램을 추가된 요구사항을 만족시키기 위해 수정하고 확장할 것을 요구하는 것이 바람직하다고 결론을 내렸다.

초보자들에게 적합한 프로그래밍 학습 전략은 잘 설계된 프로그램의 빈 부분을 채워 넣는 것으로 프로그래밍 학습을 시작하는 것이다. 다음으로 완전한 프로그램을 독립적으로 작성할 수 있도록 예제 프로그램들을 수정하고 확장하게 하는 것이다. 이와 같은 완성 전략이 초보자를 위한 더 적절한 학습 방법으로 보편화되었다. 그에 따라 대부분의 프로그래밍 입문 과목에서는 이 전략을 채택하고 있다.

2.2 프로그래밍 평가 이론

일반적으로 강의 평가는 교육 과정의 중심적 부분으로 여겨진다. 일반적인 교과목의 평가는 발달(formative) 평가와 학습 결과(summative) 평가로 이루어진다[10]. 발달 평가는 학생들의 향상 정도를 계속적으로 알아보기 위해 사용된다. 이 평가는 강의 내용을 조절하기 위해 학생들의 학습 효과를 수집하고 학생들이 실제로 배우고 있는지를 확인하고 잘못 배우고 있는 것을 알아내기 위해 사용된다. 학습 결과 평가는 성적을 주기 위한 목적으로 주어진 기간동안 배운 내용들에 대한 학생들의 지식을 확인하기 위해 강의 종료 후에 실시된다. 이 평가의 결과를 통해 강사는 자신의 강의의 전달 정도와 강의 효과에 관한 유용한 정보를 얻을 수 있다. 이러한 평가들은 보통 수량적 평가 방법을 사용하여 상대평가를 한다.

프로그래밍 과목의 평가는 프로그래밍 언어의 이론적인 내용뿐만 아니라 프로그래밍 능력을 평가해야 한다. 프로그래밍 능력을 평가하는 가장 보편적인 방법 중의 하나는 프로그래밍 과제를 통한 평가이다. 그러나 과제를 평가만으로는 충분하지 않아 시험을 통해서도 프로그래밍

능력을 평가 한다. 이 능력을 평가하기 위한 효과적인 문제 유형들의 일부는 다음과 같다[13]:

- (1) 주어진 코드의 출력 결과를 예측한다.
- (2) 주어진 문제를 푸는 코드를 작성하게 한다.
- (3) 코드의 일부분을 작성하게 한다.
- (4) 오류가 있는 코드를 수정하게 한다.

웹 기반 강의의 평가는 교육 과정의 참여자들에 의해 만들어지는 모든 정보들을 자동적으로 저장할 수 있고 언제 어디서나 평가를 하는 것이 가능하고 평가 결과를 학생들에게 빠르게 알릴 수 있다. 이러한 장점들을 활용하는 효과적인 평가 방법에 관한 연구가 이루어지고 있다. 과제물 제출은 대부분 웹상으로 이루어지고 있다. 시험은 객관식으로 실시되는 것이 보통이다. 객관식 시험은 지식의 이해나 사실들이나 개념들 사이의 관계나 사고 능력을 평가하는 데 효율적이라고 알려져 있다. 그러나 프로그래밍 능력과 같은 복잡한 성취도를 평가하기에는 적절하지 않은 면이 있다. 그러나 웹 기반 평가에서는 컴퓨터 활용 숙련도의 차이가 시험 결과에 영향을 미치고 운영 플랫폼의 제약 등의 이유로 객관식 문제나 단답형 문제 등이 사용되고 있다. 이러한 객관식 시험을 통해 프로그래밍 능력을 평가하기 위해서 위에 언급한 네 가지 문제 유형들을 객관식 문제로 만들어 출제하는 것이 한 방법이다. 좋은 객관식 문제를 출제하기 위한 제안들은 많이 발표되었다[10]. 또한 문제는 행식을 채택하는 것이 유용할 수 있다. 시험 상황에서의 변화들을 줄이기 위해 지정된 시간에 시험을 치르는 것이 권장된다.

웹 기반 평가에서 신원 확인과 부정행위가 큰 문제로 대두되고 있다[10]. 누구나 웹에 접근할 수 있기 때문에 대리 시험이 가능하고 과제물을 본인이 아닌 타인이 제출하는 것이 가능하기 때문이다. 이를 방지하기 위해 다음과 같은 방법들이 제안되었다[10]:

- (1) 강의 참여의 동기적 및 비동기적 모드사이의 일관성 확인
- (2) 강의 내용의 이해에 관한 간단하면서 자주 예고 없이 실시되는 점검
- (3) 카메라를 설치하여 본인의 사진을 전송하게 함

그러나 위의 방법들을 실제로 시행하는 것은 현실적으로 어려운 점이 있다. 이와 유사한 결과를 얻기 위한 효과적인 대안들을 찾기 위한 실제적인 연구가 필요하다.

2.3 웹 기반 강의의 운영 방안

웹 기반 강의에서는 교수와 학습자들이 직접적으로 만나지 않고 사이버 상으로 모든 교육이 이루어진다. 그러나 웹 기반 강의의 수강생들은 교육의 사회적 요소들을 경험하기 위해서 교수 혹은 다른 학생들과의 상호작용을 필요로 한다. 이러한 상호작용을 촉진시키기 위한 운영 방안은 많이 제안되었고 이를 요약하면 다음과 같다 [7,10,15]:

- (1) 인간적 요소들에 관심을 기울인다. 이를 위해 강의를 소규모로 유지한다. 학생들의 질문들, 제안 사항들, 요구 사항들에 신속하고 빠르게 응답한다.
- (2) 웹 기반 강의가 준비된 콘텐츠와 운영 플랫폼을 통해 저절로 진행되도록 내버려 두지 말고 능동적으로 강의를 이끌 필요가 있다.
- (3) 웹 기반 강의에 익숙하지 않은 수강생들을 위해 학기 초에 별도의 면대면 오리엔테이션이나 공지 사항을 통해 강의 전반에 대해 자세하게 알려 준다.
- (4) 수강생들이 매 주 일정 시간 이상 웹 기반 강의에 참여하도록 유도한다.
- (5) 학생들을 위한 상호 작용 수단을 제공하고 이를 유도하기 위한 동기 부여를 한다.

3. 연구 방법

웹 기반 강의는 90년대 중반부터 실시되기 시작하였다. 이후 웹 기반 강의 운영 사례가 보고 되고 관련 연구가 활발하게 진행되고 있다. 웹 기반 프로그래밍 강의 전략은 관련 문헌들을 조사하고 국내외에서의 사이버 강의 관련 교육을 받고 지난 4년간의 강의 경험에 기초하여 개발되었다.

교안 개발은 처음에 전통적인 면대면 강의 경험에 기초하고 기존의 사이버 강의 교안들의 문제점을 개선하기 위하여 프로그래밍 지식뿐만이 아

나라 프로그래밍 전략을 강조하여 이루어졌다. 교안을 실제 강의에 사용한 후 매 주 교안의 내용에 대한 설문 조사, 학기말에 수강생들을 대상으로 강의 전반에 대한 설문 조사, 콘텐츠 전문가에 의한 교안 평가, 여러 번에 걸친 강의 중에 수강생들로부터 받은 교안에 대한 의견등과 같은 자료들을 분석한 후 효과적인 교안 설계 전략을 도출하였다.

웹 기반 강의의 평가 전략은 국내외 강의 운영 사례, 국내외 사이버 강의 관련 교육과 실제 강의 운영 경험 등을 통해 찾아내었다. 프로그래밍 능력을 효과적으로 평가하는 방법을 찾기 위해 많은 노력을 하였다. 특히 프로그래밍 과제물의 공정하고 객관적인 평가 방법을 찾기 위해 문헌에 알려진 여러 가지 방법을 시도하였다.

웹 기반 강의의 운영 전략은 문헌들에 제시하는 방안들을 충실히 따르면 되었다. 그러나 이 방안들을 그대로 하는 것이 쉽지 않았다. 잘 훈련된 조교들을 확보하는 것이 쉽지 않았고 조교들의 사명감 있는 헌신과 노력을 유도하는 것이 어려웠다. 또한 국내 학생들의 독특한 특징이 있었다. 예를 들면 전통적인 강의와는 달리 일부 학생들이 많은 질문들, (성적 관련)요구들, 제안들, 의견들을 올렸다. 특히 이러한 요구들을 들어주지 않는 경우 자기들의 요구가 관철될 때까지 계속해서 의견을 올렸다. 반면에 일부 학생들은 강의 참여가 매우 저조하였다. 이러한 학생들은 중도에 탈락하거나 낙제 점수를 받았다. 이러한 학생들의 비율을 줄이기 위한 효과적인 방안을 찾기 위해 관련 연구 결과와 강의 운영 사례를 알아보고 실제 운영 결과에 기초하여 효과적인 운영 전략을 개발하였다.

4. 교안 설계 전략

웹 기반 교안을 개발하는 것은 기술적 관점에서는 쉽다. 그러나 교과목의 내용을 분석하고 학습자들을 고려하여 그에 따라 기획하고 설계하고 개발하는 것은 매우 어렵다. 특히 웹 기반 강의는 전통적인 면 대 면 강의와는 다른 매체, 사용자 인터페이스, 방법론을 사용하기 때문에 교과목의 특성 분석과 병행하여 교안의 사용자인 학

습자의 요구와 환경을 가능한 한 충분히 반영해야 한다.

4.1. 교안 개발 목표

개발하는 교과목이 프로그래밍 과목이므로 학습자가 교안을 학습한 후 프로그래밍 언어의 주요 개념과 이를 반영한 문법적 요소뿐만이 아니라 주어진 문제를 프로그래밍 할 수 있는 능력을 갖도록 하는 것이 첫 번째 개발 목표가 되어야 한다. 초보자가 프로그래밍을 배우는 것이 쉽지 않으므로 교안을 가능한 쉽게 개발하여 중도 탈락자나 낙제하는 학생들의 수를 줄이는 것이 두 번째 개발 목표가 되어야 한다. 인터넷을 통한 웹 기반 강의는 언제 어디서나 교육을 받을 수 있을 뿐만 아니라 필요로 하는 만큼 반복 학습이 가능하고 양방향의 대화형 통신이 가능하고 브라우저가 핵심 응용으로 자리 잡음에 따라 간단하면서도 익숙한 인터페이스를 제공하는 장점이 있어 교안 개발을 잘 하면 면 대 면 강의에 근접한 교육 효과를 기대할 수 있다. 따라서 면 대 면 강의에 근접한 교육 효과를 얻는 것이 세 번째 개발 목표가 되어야 한다.

4.2. 교안 설계 고려 사항

교안을 설계하기 전에 개발하는 교과목의 특성과 수요자인 학습자들을 고려해야 한다.

4.2.1. 프로그래밍 과목의 특성

초보자가 프로그래밍을 배우는 것은 어렵다 [5][17]. 대부분의 학생들이 프로그래밍 과목들이 매우 어렵다고 생각하고 있다. 프로그래밍 과목을 가르치는 것 또한 어렵다. 그에 따라 프로그래밍 과목은 자주 낮은 성공률과 높은 탈락률을 보이고 있다[16][17]. 이와 같이 프로그래밍을 배우고 가르치는 것이 어려운 가장 큰 이유는 프로그래밍 언어가 자연어가 아니고 인공적인 언어라는 사실이다. 그 언어는 높은 수준의 추상화를 포함한다[5]. 학생들이 프로그래밍을 할 수 있기 위해서는 그 언어의 구문론(syntax)과 의미론(semantics)을 알 필요가 있을 뿐만이 아니라 추

상적이고 논리적으로 생각할 수 있어야 한다. 매우 간단한 프로그램을 작성하기 위해서조차 학생들은 많은 추상적 개념들을 알 필요가 있다. 또한 프로그래밍 과목에는 강사가 명확하게 설명하기가 어렵거나 불가능한 상당한 양의 내재하는 지식이 있다[16][17]. 따라서 이러한 지식은 학생들에게 직접적으로 전달하기가 매우 어렵다. 학생들은 이 지식을 많은 연습을 통해 경험적으로 습득해야 한다. 그 과정에서 그들은 많은 어려움을 겪게 되고 그 결과로 흥미나 자신감을 잃게 된다. 또 다른 이유는 프로그래밍은 단순한 기술이 아니고 여러 가지의 기술들이 계층적으로 합쳐진 통합 기술이다. 프로그래밍을 잘 하기 위해서는 이러한 통합 기술을 갖추어야 한다.

4.2.2. 초보 프로그래머들의 특징

프로그래밍을 처음 배우는 초보자들의 특징은 다음과 같다[2][17]:

(1) 초보자들은 전형적으로 프로그래밍 지식과 전략들에서 부족한 점들을 많이 가지고 있다. 프로그래밍 지식을 배운 후 잊어버리거나 사용하지 않거나 부적절하게 사용한다. 특히 변수, 반복문, 배열과 회귀(recursion)등과 같은 기본적인 구문들(constructs)을 이해하는 데 많은 어려움을 겪는다[2]. 또한 주어진 문제를 프로그램할 때 계획(planning)과 설계(design)에 거의 관심을 기울이지 않는다. 또한 문제 중심 혹은 프로그래밍 중심 전략보다는 일반적인 문제 해결 전략들을 사용하여 프로그램을 작성한다. 특히 프로그램에 오류가 있는 경우 면밀한 검토보다는 대충 확인하고 부분적으로 고치려고 하는 경향이 있다.

(2) 주어진 문제에 대한 해법을 알고 있더라도 이를 프로그램으로 작성하지 못한다. 프로그래밍 언어의 각 문(statement)의 구문론과 의미론은 안다. 그러나 그 문들을 적절하게 결합하여 작동하는 프로그램으로 만드는 방법을 모른다. 특히 프로그램을 작성할 때 의미 있는 프로그램 묶음들(chunks)이나 구조들을 사용하기 보다는 문 별로 접근한다.

(3) 대부분의 초보자들은 프로그래밍을 배우고자 하는 동기와 의지가 약하다. 학생들은 모든

프로그래밍 지식을 강사들로부터 배울 수 없다. 특히 문제 해결 지식의 추상적 표현은 직접 배울 수 없고 스스로 해 봄으로써 배울 수 있다. 그들은 그 지식을 습득하기 위해서 많은 노력과 연습을 할 필요가 있다. 그러나 일부 학생들은 충분한 연습을 하지 않고 프로그래밍을 하려고 시도한 후에 프로그래밍이 잘 안되는 것에 대해 실망하고 자신감을 잃어버린다. 그리고 더 이상 노력을 하지 않고 포기하게 된다. 또한 오류가 있는 코드를 대충 검토한 후 고치려고 하고 시행착오에 의해 프로그래밍을 하는 경향이 있어 프로그래밍을 배우는 데 많은 어려움을 가진다.

4.3. 교안의 설계 방법

웹 기반 교안의 개발은 교과목의 특성 및 학습자의 요구와 환경을 고려하여 세심하게 이루어져야 한다. 위와 같은 프로그래밍 과목의 특성과 학습자들의 특징을 반영하고 프로그래밍 능력을 개발하기 위한 교안 설계 방법은 다음과 같다[14]:

(1) 주어진 문제를 프로그래밍으로 해결하는 방법을 포함시킨다.

해당 강의 주제에 적합한 문제를 프로그래밍으로 해결하는 과정을 단계적으로 자세하게 설명함으로써 문제 해결 능력을 배우게 한다. 단순히 구현 결과만을 제시하는 대신 주어진 문제의 요구 사항 분석과 이에 따른 대안들을 검토한 후에 선택한 설계에 대해 상세히 기술한다.

(2) 프로그래밍 실습 관련 내용을 포함시키고 실습 환경과 연동되게 한다.

프로그래밍을 배우는 것은 실습과 프로그래밍 연습을 통해 완성된다. 웹 기반 강의에서는 실습도 인터넷과 웹을 통해 이루어져야 하므로 교안에 실습 관련 내용이 포함되어야 한다. 특히 주어진 프로그래밍 예제 프로그램을 바로 실행시켜 결과를 확인하거나 수정할 수 있는 실습 환경과 바로 연동될 수 있어야 한다. 실습 환경은 비싼 소프트웨어를 사서 설치하는 대신에 무료로 공개된 소프트웨어를 내려 받아 설치하도록 하고 학교와 집에서 동일해야 한다. 특히 초보자가 쉽게 프로그래밍을 시작할 수 있도록 소프트웨어가 간

단하고 배우기가 쉬워야 한다.

(3) 프로그래밍 과제물을 스스로 하기 쉽게 개발하여 포함시킨다.

프로그래밍을 잘 하기 위해서는 실제로 프로그래밍을 많이 해 보아야 한다. 그러나 초보자가 교안을 학습한 후 프로그래밍을 스스로의 힘으로 하기가 쉽지 않다. 무엇보다 중요한 것이 초보자가 프로그래밍을 할 수 있다는 자신감을 가지게 하는 것이다. 이를 위해 주어진 프로그램을 그대로 작성하여 실행시키게 하는 유형으로 시작하게 하는 것이 필요하다. 다음에 주어진 잘 설계된 프로그램을 완성하고 수정하고 확장하는 유형으로 개발하여 포함시킨다. 이렇게 함으로서 초보자가 프로그램의 전체적인 구조를 알게 하고 구체적으로 해야 될 작업들이 프로그램 내에서 부분적으로 어떻게 구현되지를 배울 수 있다.

(4) 단순하고 사용하기 쉽게 개발한다.

수강생이 웹 기반 강의 교안을 학습할 때 프로그램을 많이 내려 받아 설치해야 되는 등 복잡한 일을 수행해야 한다면 수강생의 학습 의욕을 떨어뜨릴 수도 있고 최악의 경우에는 학습 진행이 이루어지지 않을 수도 있다. 또한 교안을 볼 때 도움말을 수시로 참조해야 하는 등 사용법이 복잡하다면 이 또한 학습 효과를 반감시킬 수 있다. 따라서 교안을 단순하고 사용하기 쉽게 개발하는 것이 바람직하다.

(5) 좋은 디자인을 가지게 개발한다.

수강생이 장시간 교안을 보아야 하므로 교안이 좋은 디자인을 갖도록 해야 한다. 화면 구성이 잘 정돈되어야 하고 화면마다 적절한 양의 정보가 있어야 한다. 또한 느낌이 좋고 편안한 색을 써서 눈의 피로감을 줄여 주어야 한다.

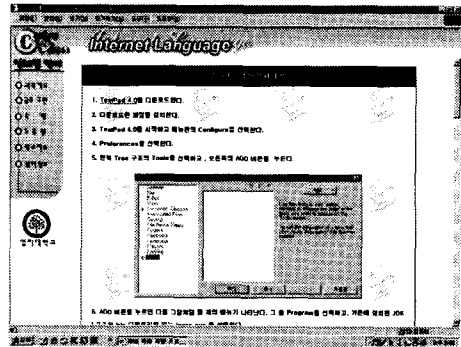
4.4 교안 개발 사례

앞 절의 교안 개발 방법에 따라 프로그래밍 기법과 실습을 강조한 프로그래밍 교안의 개발 사례를 제시한다.

4.4.1 교안의 구조

교안의 구조는 가능한 단순하게 설계한다. 교

안은 우리에게 익숙하고 네비게이션(navigation)이 쉬운 계층적 구조를 갖는다. 교안의 페이지는 학습자가 교안의 효과적인 학습을 위해 필요한 사항들인 과목 개요, 금주 교안, 숙제, 도움말, 참조 자료, 설치 정보 등을 포함한다. <그림 1>은 이러한 내용들을 포함하는 교안의 한 페이지를 보여 준다.

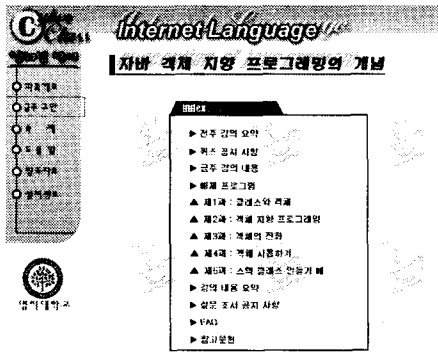


(그림 1) 실습 소프트웨어 설치 방법 설명 화면

과목 개요는 과목에 대해 전반적으로 소개하는 내용으로서 학습자가 언제든지 쉽게 참조할 수 있도록 한다. 교안의 주 내용은 주 별로 나누어 금주 교안에 포함시킨다. 숙제는 해당 주의 강의 내용에 대한 프로그래밍 과제를 포함한다. 도움말은 교안을 보는 것을 도와주기 위해 교안의 메뉴와 아이콘에 대해 설명한다. 참조 자료는 프로그래밍에 필수적인 자바 클래스 라이브러리의 API를 필요시 참조할 수 있도록 해당 웹 페이지에 연결되는 연결점(link)을 포함한다. 설치 정보는 학습자가 프로그래밍 실습 환경을 설치하는 과정을 자세히 설명한다. 먼저 자바 클래스 라이브러리인 JDK를 설치하는 방법을 설명하고 다음으로 공개된 간단한 자바 프로그래밍 소프트웨어인 TextPad 설치 방법을 설명한다. <그림 1>는 실습 소프트웨어인 TextPad를 설치하는 방법을 설명하는 내용을 캡처한 화면이다.

교안의 주 내용인 금주 교안은 면 대 면 강의의 전형적인 순서를 따르고 프로그래밍 과목의 특성을 반영하여 예제 프로그램을 포함시킨다. 그 밖에 전주 강의 내용에 대한 퀴즈와 교안 학습 후의 교안 평가를 위한 설문 조사는 웹 기반 강의 운영 플랫폼에서 제공하는 기능을 활용하므로 이를 알리는 공지 사항을 포함한다. 금주 교

안에는 전주 강의 요약, 퀴즈 공지 사항, 금주 강의 내용, 예제 프로그램, 과로 나뉘어진 소주제들, 강의 내용 요약, 설문 조사 공지 사항, FAQ와 참고 문헌등이 있다. FAQ는 강의 내용 중 예상되는 질문들을 모아 놓은 것이다. 참고 문헌은 강의 내용과 관련된 자료 등을 포함한다. <그림 2>는 이러한 내용을 보여 주는 한 주 교안의 초기 화면으로서 왼쪽 메뉴의 금주 교안을 누르면 나온다.

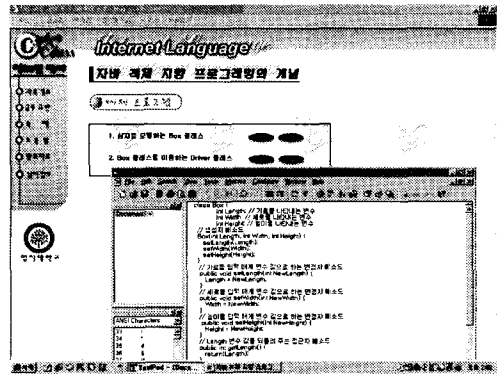


(그림 2) 한 주 교안의 초기 화면

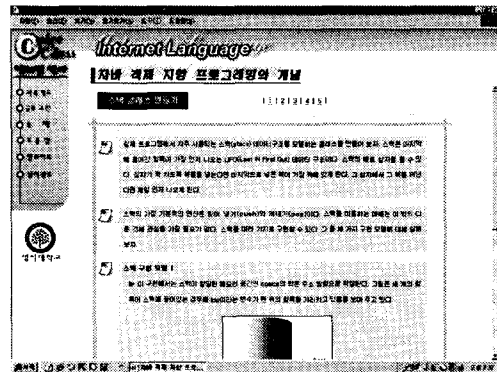
예제 프로그램은 해당 주의 교안 내용을 이해하면 읽을 수 있고 작성할 수 있는 대표적인 프로그램의 코드와 실행 결과를 보여 준다. 제시된 프로그램에 대해 보기(view) 기능과 내려 받기(down) 기능이 있다. 내려 받기 기능은 프로그램을 내려 받아 현재 위치에서 연동된 자바 개발 소프트웨어인 TextPad를 통해 열어 그대로 혹은 수정하여 컴파일하고 실행해 볼 수 있도록 한다. <그림 3>은 한 주 교안의 예제 프로그램을 내려 받아 현재 위치에서 TextPad를 통해 열린 화면을 보여 준다. 따라서 학습자는 교안을 학습하는 중에 주어진 프로그램을 가지고 실습을 할 수 있다.

주어진 문제를 프로그램하는 방법을 단계별로 자세하게 설명한다. 특히 프로그램 설계 부분에 중점을 두어 여러 대안들을 제시하고 그 중 하나를 선택하는 이유를 제시한다. 다음으로 선택된 프로그램 설계에 따라 어떻게 구현하는지를 상세하게 설명한다. <그림 4>은 대표적인 데이터 구조 중의 하나인 스택을 모델하는 클래스를 프로그램하는 단계 중 문제 분석과 설계 대안들을 고려하는 과정을 보여 준다. <그림 5>은 대안들 중

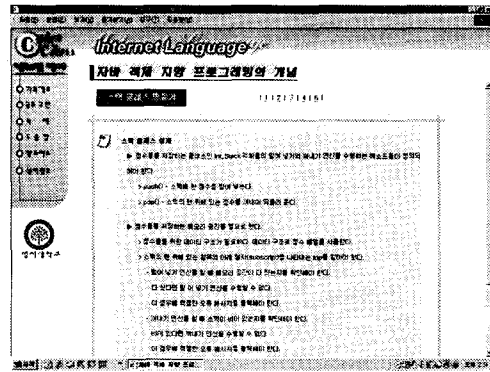
선택된 프로그램 설계를 보여 준다. <그림 6>는 설계에 따라 구현된 프로그램을 보여 준다.



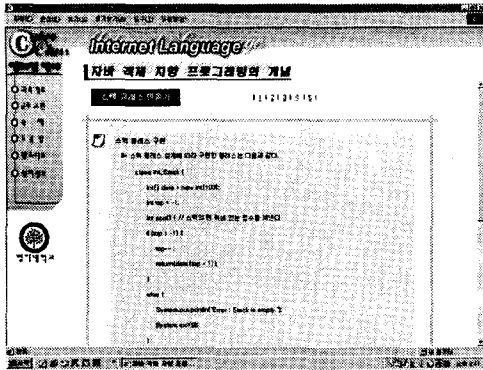
(그림 3) 예제 프로그램의 내려 받기 화면



(그림 4) 문제 분석 및 설계 대안 제시 화면



(그림 5) 선택된 프로그램 설계 화면



(그림 6) 설계에 따라 구현된 프로그램 화면

4.4.2 교안의 화면 구성

교안의 화면 구성은 가능한 단순하게 설계한다. 이렇게 함으로서 학습자가 교안 사용법을 배우지 않고도 쉽게 교안을 볼 수 있게 한다. 교안의 화면은 크게 세 부분으로 나눈다. 윗 부분은 표제 영역으로서 사이버 강의란 사실과 교과과목명 등을 알리는 고정된 텍스트가 있다. 왼쪽 부분은 메뉴 영역으로서 고정된 교안의 메뉴와 로고가 있다. 메뉴는 과목 개요, 금주 교안, 숙제, 도움말, 참조 자료와 설치 정보와 같은 6 개의 메뉴 항목들을 포함한다. 이는 <그림 1>의 왼쪽에 보여져 있다. 오른쪽 부분은 변경될 수 있는 주 영역으로서 화면의 대부분을 차지하고 왼쪽에 있는 특정 메뉴 항목을 누르면 그에 해당하는 내용이 나온다. 예를 들면 <그림 2>에 보여진 대로 금주 교안 메뉴를 누르면 오른쪽 화면에 1주 분의 강의 내용이 나온다. 주 영역에 있는 특정 항목을 누르면 그 영역이 그 항목에 대한 내용으로 교체되거나 별도의 창이나 대화창이 뜬다. <그림 2>의 화면에서 '제 5 과: 스택 클래스 만들기'를 누르면 <그림 4>의 화면이 나온다. 교안의 모든 내용은 대부분의 경우 마우스를 3 번 누르면 볼 수 있고 최대 4 번 누르면 볼 수 있다.

화면 구성을 일관성이 있고 정돈되게 하여 단기간 내에 학습자가 교안의 구성에 익숙하게 한다. 또한 화면 공간을 넉넉하게 사용하도록 설계한다. 모든 화면에 충분한 여백을 두어 학습자가 지루함을 느끼지 않고 여유를 가지고 교안을 볼

수 있도록 한다. 또한 느낌이 좋고 편안한 색을 사용하여 좋은 기분으로 학습하게 한다. <그림 4>는 이러한 화면의 한 좋은 예이다.

5. 강의 평가 및 운영 전략

이 장에서는 관련 문헌들과 수년 간에 걸쳐 이루어진 웹 기반 프로그래밍 강의 경험에 기초하여 효과적이고 실제적인 강의 평가 및 운영 전략을 제시한다. 웹 기반 강의의 어려운 과제중의 하나가 효과적인 평가 방법을 고안하는 것이다. 이는 웹 기반 강의의 성공을 위한 핵심 요인 중의 하나이다. 또한 웹 기반 강의의 수강생들은 교수 혹은 다른 학생들과의 상호작용을 필요로 하고 실제로 상호작용을 적극적으로 추구한다. 이러한 상호작용을 촉진시키기 위한 운영 방안이 마련되어야 한다.

5.1. 평가 전략

5.1.1 과제 평가 전략

프로그래밍 과목에서 프로그래밍 능력을 배우기 위해서는 과제가 가장 중요하다. 과제는 프로그래밍을 연습하는 유형이어야 한다. 초보자가 프로그래밍을 배우기가 어려우므로 프로그램을 쉽게 할 수 있는 과제를 부과하는 것이 적절하다. 해당 주의 강의 내용을 이해하면 프로그램을 할 수 있는 문제를 선택해야 한다. 가능한 작은 크기의 쉬운 프로그램 과제를 매주 부과하는 것이 좋은 방법 중의 하나이다.

과제는 풀어야 할 문제에 대한 미완성인 프로그램의 비어있는 곳을 채우는 유형이 적절해 보인다. 이 유형은 완성 전략을 따른 것으로서 학생들이 풀어야 할 문제에 집중하게 한다. 이 전략은 초보자가 문제 해결 능력을 배우는 유용한 방법 중의 하나이다. 미완성 부분을 채우는 것을 도와주기 위해 문제와 미완성 프로그램 코드의 각 부분에 대한 자세한 설명을 담고 있는 정보 파일을 만드는 것이 적절하다. 학습자가 이 정보 파일을 읽으면 교안에서 배운 강의 내용이 실제 문제를 프로그래밍 하는데 어떻게

적용되는지를 알 수 있고 미완성 부분도 채워 넣을 수 있도록 한다. 또한 학습자가 강의 내용 관련 부분에 집중하도록 프로그램 내의 입출력 부분은 완성시킨다. 이렇게 함으로써 학습자가 프로그램을 정확하게 완성하면 출력 결과는 문제의 정답과 일치해야 한다. 따라서 학습자가 완성한 프로그램의 정확성을 스스로 확인할 수 있고 자동 채점 프로그램을 개발하여 채점할 수 있다. 주어진 프로그램은 바로 검토하고 실행할 수 있어야 한다.

미완성 프로그램의 빈 곳을 채우는 유형의 과제물은 초보자가 쉽게 프로그래밍 할 수 있게 하지만 남의 프로그램을 베끼는 부정행위가 있을 경우 적발하기가 어려운 단점이 있다. 실제로 이러한 유형의 과제물을 부과한 결과 적지 않은 학생들의 과제물이 거의 같았고 평가를 공정하고 합리적으로 하기가 어려웠다. 그 결과 과제물의 평가 결과가 변별력이 없었다. 이 문제를 해결하기 위한 방안을 마련하는 것이 필수 불가결하다.

강의 후반부에 풀어야 할 문제를 주고 프로그램을 작성하는 유형의 과제물을 부과하는 것은 좋은 방안 중의 하나이다. 이 과제물들은 한 프로그램을 여러 단계를 거쳐 완성하는 유형으로 하는 것이 바람직하다. 즉, 프로그램들이 서로 연계되어 있어 한 프로그램을 완성한 다음에 그 프로그램을 확장하여 다음 프로그램을 완성하는 형태이다. 새로운 유형의 과제물은 학생들에게 어려울 수 있다. 이를 위해 문제 요구사항, 힌트 등을 공지 사항으로 올리는 등의 노력이 필요하다. 일반적으로 학생들은 직접 프로그래밍을 해 보고 어려운 점에 대한 피드백을 받고 그 후 생각함으로써 배운다. 따라서 프로그래밍을 배우는 데 빠르고 자주 이루어지는 피드백이 매우 중요하다.

5.1.2 시험 평가 전략

시험 평가는 웹 기반 강의의 성공적인 운영을 위해 주의 깊게 계획되고 실시되어야 한다. 특히 프로그래밍 과목의 교육 목표 중의 하나가 프로그래밍 능력을 습득하는 것이기 때문에 시험 평가에서도 이 능력을 평가해야 한다. 시험 평가는 수시 시험, 중간 시험, 기말 시험으로 이루어지는 것이 적절한 방안이다. 수시 시험은 전주 강의 내용에 대해 매주 실시하여 복습을 하게 하고 매주 교안을 습관적으로 학

습하도록 유도하기 위한 목적이다. 시험 문제는 이론적인 내용뿐만 아니라 프로그래밍 능력을 평가할 수 있도록 설계되어야 한다.

시험은 보통 사이버 강의 운영 플랫폼을 사용하여 실시되므로 기능 제약에 따라 객관식 유형의 문제만이 제공되는 것이 보통이다. 객관식으로 프로그래밍 능력을 평가해야 함으로 문제 출제에 각별한 노력이 필요하다. 시험 문제들을 문제 은행식으로 준비해 놓고 이 문제들 중 임의로 선택하여 출제하는 것도 좋은 방법 중의 하나이다.

시험이 사이버 운영 플랫폼 상에서 실시되어 학생들이 익숙하지 않을 수 있으므로 시험 보기 몇 주 전부터 시험 일정, 출제 방향, 공부 내용, 시험 보는 방법 등과 같은 시험 정보를 공지사항을 통해 여러 번에 걸쳐 알려야 한다. 시험은 공정성과 부정행위 방지를 위해 모든 학생들이 같은 시간에 보도록 하는 것이 바람직하다.

5.2. 강의 운영 전략

웹 기반 강의에 익숙하지 않은 수강생들을 위해 학기 초에 별도의 면 대 면 오리엔테이션이나 공지사항을 통해 교수요목, 수강 방법, 학습 방법, 성적 평가 방법 등을 알려 주는 것이 강의의 원활한 운영에 필수 불가결하다. 특히 수강생들이 면 대 면 강의보다 수강하기가 쉽다는 선입관을 바꿔주기 위해 학기 초부터 강의에 적극적으로 참여하도록 유도할 필요가 있다. 실제로 강의 시작 후 몇 주가 지나도 강의 운영시스템을 통해 교안을 보거나 과제물을 내려받거나 올리는 등과 같은 기본적인 사항들에 익숙하지 않아 어려움을 겪는 수강생들이 있다.

웹 기반 강의가 준비된 콘텐츠와 운영 플랫폼을 통해 저절로 진행되도록 내버려 두지 말고 능동적으로 강의를 이끌 필요가 있다. 특히 강사와 조교가 수강생들에게 지속적으로 관심을 가지고 있고 강의 운영에 적극적으로 참여하고 있다는 것을 알릴 필요가 있다. 이를 위한 좋은 방법 중의 하나는 수강생들의 학습 의욕을 고취시키고 학습에 직접적인 도움을 주기 위해 질의응답과 게시판에 올라오는 글에 대한 신속한 응답(게시 후 2일 이내)을 하는 것이다. 특히 과제물과 관련된 질문에 대한 신속한 응답이 필요하다. 실제로 사이버 강의에서는 면 대 면 강의에서 전

혀 경험해 보지 못한 활발한 질의와 의견 제시 등이 이루어진다. 이러한 글들에 대한 빠른 응답을 통해 강사와 조교가 항상 가까이에서 도움을 주려고 노력하고 있다는 사실을 느끼게 하는 것이 강의의 원활한 진행에 필수 불가결하다. 응답이 늦어지면 불만의 글이 계속 올라오면서 생산적인 상호작용의 장이 마련되지 않고 부정적이고 비판적인 상호작용의 장이 형성된다. 이러한 일이 일어나지 않도록 각별한 노력이 필요하다.

수강생들이 매 주 일정 시간 이상 웹 기반 강의에 참여하도록 하는 시스템의 구축이 필요하다. 매 주 교안을 본 시간을 평가에 반영하거나 전 주에 배운 내용을 다음 주에 수시 시험을 보도록 한다거나 매 주 과제물을 제출하게 하는 것이 한 방안이 될 수 있다. 다만 학습량이 과다하지 않도록 난이도를 조절할 필요가 있다. 수시 시험과 같은 매 주 실시하는 평가 요소에 대해 꾸준히 공부하게 하고 참여도를 높이기 위해 배점을 높일 필요가 있다.

웹 기반 강의의 수강 인원의 수를 제한하는 것이 성공적인 강의 운영을 위해 꼭 필요하다. 실제로 질의응답/게시판에 올라오는 질문들, 의견들이나 비판들을 읽고 답하고 과제물을 수행하도록 도와주고 수시 시험/시험/과제물의 정답과 채점 결과를 공개하고 채점에 대한 이의 제기에 응답하는 일은 강사와 조교에게 많은 노력과 시간과 인내심을 요구한다. 따라서 질적이고 효율적인 강의 운영을 위해서 소규모의 강의가 필수적이다.

웹 기반 강의에서는 교수와 학습자들이 먼 대면으로 직접 만나 의사소통할 수가 없으므로 학생들을 위한 상호 작용 수단을 제공하고 이를 유도하기 위한 동기 부여를 하는 것이 매우 중요하다. 수강생들이 강의에 적극적으로 참여하도록 하기 내내 노력을 할 필요가 있다. 수강생들이 올린 글들에 대해 성실하게 답하고 어려움이 있을 때 도움을 요청하라고 촉진하고 수강생들에게 기대하는 것을 알리고 건의 사항들을 올리게 하고 강의를 재미있게 운영하는 등의 노력이 필요하다. 수시로 칭찬과 격려를 하고 강의를 성공적으로 이수한 후의 이점들을 알리는 등의 동기 부여를 하는 공지 사항을 올리는 것도 효과적으로 보인다.

6. 강의 운영 사례

6.1 강의 운영 환경

자바를 다루는 인터넷 언어라는 과목의 웹 기반 강의를 지난 4년간 8회에 걸쳐 본 대학 전자정보통신공학부 학생들과 정보통신 사이버대학 협의회에 소속된 대학의 학생들을 대상으로 진행되었다. 이 강의는 정보통신 사이버대학 협의회에 소속된 9개 대학의 38명의 학생들을 대상으로 전 과정을 웹 기반으로 진행하였다. 수강생들의 전공은 대부분 정보기술 분야이고 기타 전공이 9명이다.

강의 과목의 성적 평가 요소는 출석, 강의 참여, 과제, 수시 시험, 중간/기말 시험이다. 출석은 강의 운영시스템의 이용 시간을 기준으로 하였으며 매 주 3시간 이상씩 이용해야 출석으로 인정하였다. 강의 참여는 질의응답에 질문(혹은 응답)을 올리거나 자유게시판에 글을 올린 횟수로 평가하였다. 과제는 강의 전반부에는 완성 전략을 따르는 주어진 미완성 프로그램을 완성하는 유형이고 강의 후반부에는 생성 전략을 따르는 주어진 문제를 프로그래밍하는 유형이다. 과제물은 1~2 주마다 한 번씩 총 10번 부과되었다. 수시 시험은 전 주에 배운 내용에 대해 매 주 수강생들이 제한된 시간 내에 각자 편리한 시간에 치르는 형식으로 총 12회 실시되었다. 중간시험과 기말 시험은 강의운영시스템의 기능 제한에 따라 각각 학기 중간과 말에 객관식으로 정해진 시간에 실시되었다. 각 평가요소의 반영률은 출석 10%, 강의 참여 8%, 과제 30%, 시험 52%(수시시험 12%, 중간 시험 20%, 기말 시험 20%)이다.

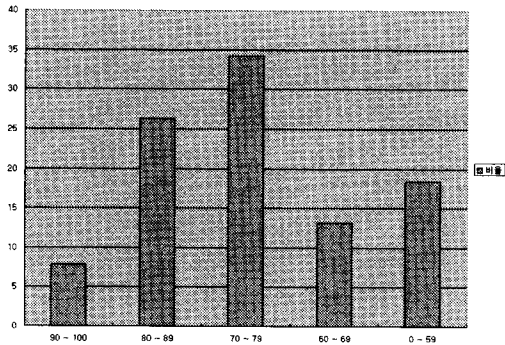
6.2 성적 평가 결과 및 고찰

수강생들의 최종 성적과 성적 분포는 각각 <표 1>과 <그림 7>과 같다. 성적의 특징은 표준 편차가 상대적으로 크고 평균이 중앙값에 비해 매우 낮다는 것이다. 이는 <그림 7>의 성적 분포가 잘 보여 주고 있다. 수강생들의 점수가 대부분 중상위권(78점 이상)에 분포되어 있는 데 비해 낙제 점수(60점 미만)를 받은 학생들의 수가 상대적으로 많고 그들의 점

수가 매우 낮다. 낙제한 학생들이 비율(18.4%)은 이전 강의들에서는 30%가 넘었으나 많이 낮아 졌다.

<표 1> 수강생들의 성적

통계량	값
평균	70.5
표준 편차	17.8
중앙값	77.9
최고 점수	91.8
최저 점수	19.2
평균 절대 편차	12.0
사분위수범위	15.2
수강생수	38



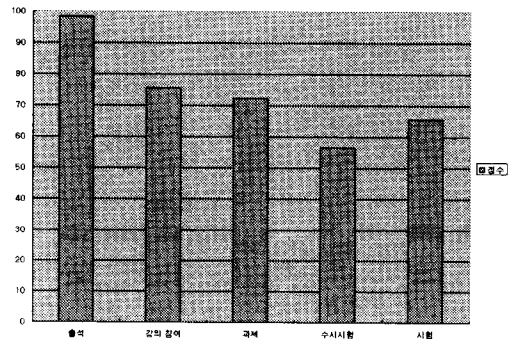
(그림 7) 수강생 성적 분포

평가 요소별 점수는 <그림 8>와 같다. 출석 점수는 매우 좋고 강의 참여와 과제 점수는 비교적 좋다. 그러나 수시 시험과 (중간/기말) 시험에서 낮은 점수를 보이고 있다. 출석 점수는 대부분의 수강생들이 운영시스템을 평균적으로 3시간 이상 보아서 거의 만점에 가깝다. 매 주 강의 내용을 그 다음 주에 수시 시험에서 평가한 것이 교안 학습을 규칙적으로 일정 시간 이상 하게 한 것으로 보인다. 강의 참여 점수도 많은 학생들이 질의응답이나 게시판에 많은 글을 올리는 등 강의에 적극적으로 참여하여 좋은 편이다.

과제 점수는 낮은 편이다. 각 과제물의 제출 비율과 평균 점수는 <표 2>와 같다. 과제물의 제출 비율과 평균 점수는 강의가 진행될수록 어려워져서 낮아 졌다. 특히 강의 후반부 (6번째에서 10번째)의 과제 점수가 강의 전반부(1번째에서 5번째)의 점수보다 매우 낮음을 확인할 수 있다.

이는 강의 전반부에는 과제가 일부 비어있는 프로그램 부분을 완성시키는 형태라 비교적 쉬웠으나 강의 후반부의 과제는 문제를 주고 프로그램을 작성하는 유형으로 수강생들에게 다소 어려웠고 강의를 충실히 듣지 않은 수강생들은 과제를 제출하지 않았기 때문에 판단된다.

학생들의 적극적인 수업 참여를 위해 매주 수시 시험을 보았다. 수시 시험의 경우는 점수가 매우 낮았고 강의 전반부와 후반부에 차이가 거의 없었다. 이는 자바 언어의 문법적 요소와 프로그래밍 개념들뿐만이 아니라 프로그래밍 능력을 평가하는 문제들이 학생들에게 어려웠기 때문에 판단된다. 특히 중간/기말 시험 기간 전에 성적이 낮았다. 이는 다른 과목들의 시험 준비를 하느라 강의 수강을 소홀히 했기 때문으로 추정된다. 또한 중간/시험 점수도 기대한 것보다 낮게 나왔다. 이는 프로그래밍 능력을 평가하는 문제가 어려웠기 때문으로 사료된다.



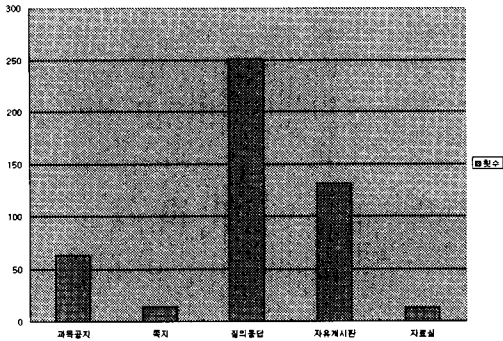
(그림 8) 평가요소별 점수

<표 2> 과제 제출 결과

과제번호	제출 비율	평균점수
1	95.3	91
2	93.4	88
3	92.5	82
4	88.7	72
5	84.1	74
6	79.4	43
7	70.9	45
8	63.5	53
9	61.6	51
10	59.8	52

6.3 강의 운영 결과 및 고찰

웹 기반 강의에서 수강생과의 상호작용을 위한 운영 요소는 과목 공지, 쪽지, 질의 응답, 자유게시판, 자료실 등이 있다. 각 운영 요소의 이용 횟수는 <그림 9>과 같다. 수강생들이 강의 운영 시스템에 접속한 총 시간은 약 2,843시간이며, 한 사람당 매 주 평균 4.67시간을 접속했다. 이는 학생들의 사이버 강의에 대한 적극적인 참여를 보여 주고 있다.



(그림 9) 각 운영 요소의 이용 횟수

과목 공지에는 평균적으로 주당 4개 총 63개의 글이 게시되었다. 게시물의 주 내용은 앞으로의 학습 방향과 학습을 권하는 글을 포함해 성적 평가 방법과 숙제에 대한 힌트, 중간/기말시험 계획 및 일정, 과제 유형의 변경, 성적 처리 기준, 시험 채점 결과, 부정행위에 대한 경고 등이다. 또한 학기 중반 이후에는 질의응답과 자유게시판에 생산적이고 합리적인 내용의 글을 올릴 것을 권유하는 글 등을 올렸다. 그 결과 질의응답이나 자유게시판을 통해 적극적인 강의 참여와 수업에 대한 열정을 보여 주었다. 쪽지는 강의 수강을 소홀히 하거나 과제물을 제출하지 않는 학생들에게 수업 참여를 독려하기 위해 총 14회 사용되었다.

질의응답의 경우 강의가 끝날 때까지 총 251개의 게시물이 올라왔다. 따라서 수강생은 평균적으로 6.6개의 게시물을 올린 셈이다. 게시물의 총 수는 게시물에 대한 응답을 포함하면 360개 이상이 될 것이라고 추정된다. 그 만큼 많은 학생들

의 적극적인 참여와 수업에 대한 깊은 관심을 확인할 수 있었다. 게시물의 내용들을 보면 강의 초반에는 수강 및 과제 제출 방법에 대한 질문과 강의 교안을 볼 때 나타나는 오류에 대한 질문이 주류를 이루었다. 중반에 오면 대부분이 과제에 대한 질문과 수시 시험에 대한 질문이 주류를 이루었다. 많은 수강생들이 과제를 수행하는데 어려움을 겪었다. 주로 과제에서 요구하는 것을 이해하지 못하거나 어떻게 해야 할 지를 묻는 것들이었다. 수시 시험에 대한 질문은 강의 초에는 시험이 제대로 종료되었는지를 확인하는 질문이 많았다. 후반부에는 과제와 수시 시험에 대한 질문들과 함께 과제와 시험 성적에 대한 문의의 비중이 많았다.

자유게시판의 경우 강의 종료 시까지 총 131개의 게시물이 올라왔다. 게시물의 주 내용은 강의 초반에는 수업에 대한 적극적 관심을 표명하는 글들이 많았다. 중반에는 과제와 수시시험의 어려움을 호소하는 사람들이 많았다. 또한 앞으로 더 열심히 해야겠다는 수업에 대한 적극적인 참여 의지를 표현하는 글들도 있었다. 중반에는 강사와 조교들에 대한 감사의 글들이 있었고 더 열심히 하지 못해 아쉽다는 글들도 많았다. 자료실에는 총 13개의 글과 함께 자료가 게시되었다. 과제에 대한 모범 답안과 기 출제된 중간시험 문제들과 도움말 파일 등을 올려놓았다.

7. 결 론

본 논문은 웹 기반의 프로그래밍 교안을 개발한 후 이 교안을 사용하여 수년에 걸쳐 강의를 운영한 경험과 관련 문헌들에 기초하여 웹 기반 프로그래밍 과목의 효과적인 강의 전략을 제시하였다. 이러한 전략에 따라 최근에 운영된 강의의 운영 사례를 요약하고 분석하였다. 이 결과는 강의 운영을 잘하면 웹 기반 강의의 교육 효과를 높일 수 있음을 보여 준다. 특히 이전의 웹 기반 강의의 가장 큰 문제점 중의 하나였던 중도에 포기하는 학생들과 낙제점을 받는 학생들의 수를 많이 줄일 수 있었다. 향후 프로그래밍 과목을 계획하고 있는 강사들에게 하나의 참고 사례로서 활용되어 사이버 강의의 효율적인 운영에 기여하기를 기대한다.

참 고 문 헌

[1] L.E. Deimel and D.V. Moffat(1982). A more analytical approach to teaching the introductory programming course. *Proceedings of the NECC*.

[2] E. Soloway and J.C. Spohrer(1989). Studying the novice programmer, Lawrence Erlbaum.

[3] J.J.G. van Merrinboer(1990). Strategies for programming instruction in high school: Program completion vs program generation. *J. Educational Computing Research*, 6(3), pp. 265-285.

[4] R. Oliver(1993), Measuring hierarchical levels of programming knowledge. *J. Educational Computing Research*, 9(3), pp. 299-312.

[5] R. Moser(1997). A fantasy adventure game as a learning environment: why learning to program is so difficult and what can be done about it. *Proceedings of the Annual Joint Conference Integrating Technology into Computer Science Education*, Uppsala, Sweden.

[6] F.P. Deek and J.A. McHugh(1998). A Survey and Critical Analysis of Tools for Learning Programming. *Computer Science Education*, 8(2), pp. 130-178.

[7] C. McCormack and D. Jones(1998). Building a Web-based Education System, John Wiley & Sons.

[8] 이종연(1999). 사례: 삼성 사이버 교육 현황 및 발전 방향. *정보과학회지*, 17(9), pp. 47-51.

[9] A. Ausserhofer(1999). Web-Based Teaching and Learning: A Panacea?. *IEEE Communications Magazine*, 3, pp. 92-96.

[10] A. Aggarwal(2000). Web-Based Learning and Teaching: Opportunities and Challenges, Idea Group Publishing.

[11] K. Chang, B. Chiao, S. Chen, and R. Hsiao(2000). A Programming Learning System for Beginners - A Completion Strategy. *IEEE Transactions on Education*, 43(2), pp. 211-220.

[12] 이충기(2001). 자바 프로그래밍 교육에서 웹 기반 강의와 면 대 면 강의의 학습 효과 비교 분석 연구. *공학교육연구*, 4(2), pp. 3-10.

[13] M.J. Van Gorp and S. Grissom(2001). An

Empirical Evaluation of Using Constructive Classroom Activities to Teach Introductory Programming. *Computer Science education*, 11(3), pp. 247-260.

[14] 이충기(2002). 웹 기반의 자바 프로그래밍 강의 교안 개발. *공학교육연구*, 5(1), pp. 3-18.

[15] M. Khosrow-Pour(2002). *Web-Based Instructional Learning*, IRM Press.

[16] N. Truong, P. Bancroft, and P. Roe(2002). ELP - A Web Environment for Learning to Program. *Proceedings of the ASCILITE 2002 Conference*.

[17] A. Robins, J. Rountree, and N. Rountree(2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), pp. 137-172.



이 충 기

1979 서울대학교 계산통계학과 (이학사)
1981 서울대학교 계산통계학과 (이학석사)

1993 조지아공과대학교 전산대학 (전산학박사)
1994~1996 한국전산원 표준본부 선임연구원
1996~현재 명지대학교 컴퓨터소프트웨어학과 교수

관심분야: 웹기반교육, 이동 네트워킹

E-Mail: cklee@mju.ac.kr



홍 석 원

1979 서울대학교 물리학과 (이학사)
1988 노스캐롤리나주립대학교 전산학과(이학석사)

1992 노스캐롤리나주립대학교 전산학과 (전산학박사)

1993~1995 한국전자통신연구원 선임연구원
1995~현재 명지대학교 컴퓨터소프트웨어학과 교수

관심분야: 컴퓨터 네트워킹

E-mail: swhong@mju.ac.kr