

대학 교양교육으로서의 컴퓨팅사고와 프로그래밍

연세대학교 학부대학 | 김재경

1. 서 론

2016년 1월 스위스에서 개최된 다보스 포럼에서는 4차 산업혁명을 핵심 주제로 다루며 변화하는 미래 사회에 어떻게 대응하고 적응할 것인가에 대한 논의가 이루어졌으며[1], 2017년 가트너 그룹에서도 인공지능, 몰입경험 환경 및 대화형 서비스 등이 미래 사회를 주도할 핵심 기술로 예측하였다[2]. 이와 같이 급변하는 산업 구조에 대비하여 국내외에서는 소프트웨어 교육을 통한 우수한 인재 양성을 하는 것이 최우선 과제가 되었다.

소프트웨어 교육의 중요성에 따라 교육부에서는 2015 개정 교육과정을 발표하여 초등학교는 2019년부터 17시간, 중학교는 2018년부터 단계적으로 34시간 이상 소프트웨어 교육을 필수화하여 중등교육에서는 이미 올해부터 소프트웨어 교육이 시행되고 있다[3].

또한 교육부와 미래창조과학부(현 과학기술정보통신부)에서는 ‘소프트웨어 교육 활성화 기본계획’을 2016년도에 심의하여 확정하였다. 먼저 교원 확충과 기반 인프라를 구축하여 초·중등 소프트웨어 교육을 활성화하고, 소프트웨어 중심대학 지원사업을 통하여 고등교육에서 비전공자의 소프트웨어 기초 교육을 의무화하는 등 교육의

혁신을 추구하고 있다[4].

소프트웨어 인재양성은 기존의 제조업과 다르게 설비 투자 비용이 적게 들지만 고도의 지식 집약 산업으로 창의·융합적 사고력과 컴퓨팅 사고를 통한 문제 해결 능력을 가진 인적자원을 양성하는 것이 핵심이다[5]. 이와 같은 특성에 따라 소프트웨어 인재 양성은 단기간에 해결할 수 있는 것이 아니며 초·중등 및 고등 교육과정에서 지속적이고 체계적인 교과과정을 수립하여야 한다.

특히 교과 개발 과정에서는 학습자들이 소프트웨어 교육에 대해 기대하는 요소들과 소프트웨어를 학습하려는 이유와 목적에 대해서 파악 및 분석하여 이에 맞는 실용적인 교과를 구성하는 것이 요구된다.

이와 같은 요구를 반영한 소프트웨어 교과목을 통해 비전공자가 컴퓨팅사고를 바탕으로 문제 해결을 할 수 있도록 하고, 나아가 다양한 프로그래밍 실습을 자신의 전공 분야에 응용할 수 있도록 기반을 마련하여 주어야 한다.

또한 학습자들의 피드백을 분석하고 개선점을 지속적으로 반영하여, 다양한 배경을 가진 비전공자들이 소프트웨어 교육을 어려움 없이 성공적으로 받을 수 있도록 환류체계를 개선하고 관리하는 것이 중요하다.

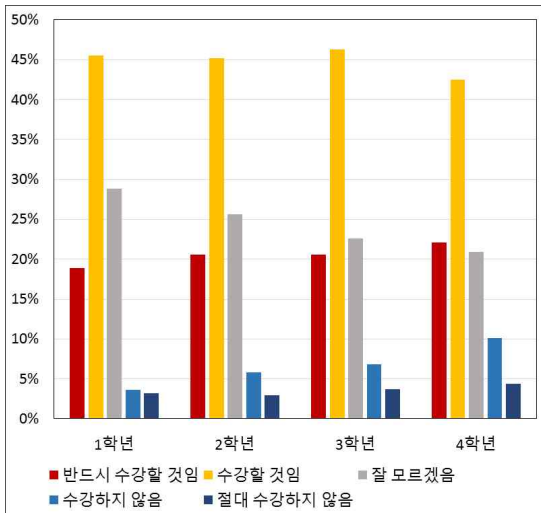
2. 소프트웨어 교양 과목의 개설

2.1 소프트웨어 과목에 대한 요구사항 조사

연세대학교에서는 2017년 필수교양 과정에 SW 영역을 신설하고 전교생을 대상으로 소프트웨어 교양 교육을 시행하기에 앞서 재학생을 대상으로 과목 개설에 대한 설문을 진행하였다.

이를 통하여 학생들의 소프트웨어 교과에 대한 수요와 기대하는 교육 효과 및 수강 이유 등을 조사하고 이에 맞는 교과를 구성하기 위해서이다.

전체 재학생인 11,297명을 대상으로 2016년 10월에 설문을 실시하였으며 응답자는 3,532명으로 31.3%의 학생이 설문에 응하였다[6]. 교내 온라인 수업 관리 시스템[7]에 설문을 게시하여 진행하였고 강제사항이나 설문에 대한 별다른 보상이 없었음을 감안하면 응답률로부터 소프트웨어 과목 개설에 대한 학생들의 관심이 매우 높았음을 알 수 있었다.



소프트웨어 과목을 개설할 경우 수강 의향을 묻는 질문에 긍정적인 답변을 한 학생은 총 2,306

명(65.3%)의 비율을 차지하였다. 이 중 721명(31.3%)의 학생은 ‘반드시 수강할 것임’의 응답을 하였다. 학생들도 4차 산업혁명의 변화에 대비하여 새로운 분야에 도전하려는 의지가 높음을 확인할 수 있었다.

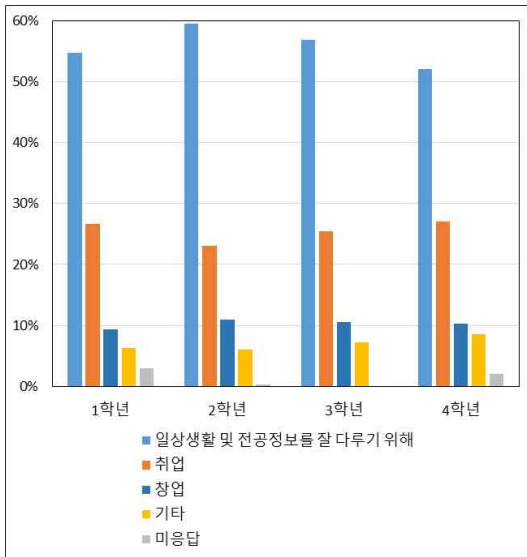
<SW 과목 개설 시 수강 의향에 대한 응답인원>

학년별	응답	반드시 수강할 것임	수강할 것임	잘 모르겠음	수강하지 않음	절대 수강하지 않음	계
1학년	인원	211	509	322	40	36	1,118
	비율(%)	18.9	45.5	28.8	3.6	3.2	31.7
2학년	인원	164	361	204	46	23	798
	비율(%)	20.6	45.2	25.6	5.8	2.9	22.6
3학년	인원	154	346	169	51	28	748
	비율(%)	20.6	46.3	22.6	6.8	3.7	21.2
4학년	인원	192	369	181	88	38	868
	비율(%)	22.1	42.5	20.9	10.1	4.4	24.6
계		721	1,585	876	225	125	3,532

반대로 350명(9.9%)의 학생이 부정적인 응답을 하여 거부감을 나타내었다. 비록 긍정적인 응답에 비해 매우 적은 수치이나 새로운 산업과 기술에 대해 다소 보수적인 학생들도 존재하였다. 학년별로는 큰 차이를 보이지 않았으나 그림에서 보듯이 4학년 학생들이 부정적인 답변의 비율이 조금 높았다. 취업을 앞두고 변화하는 산업과 기술에 부담을 느끼는 것으로 추측된다.

설문 결과를 통해 학생들 역시 소프트웨어 교육에 매우 높은 관심과 요구가 있음을 알 수 있었으며 관련 교과목 개설이 필요함을 확인하였다.

다음으로 프로그래밍을 배우고 싶다면 그 이유는 무엇인가에 대해 설문을 실시하였다. 가장 많은 대답은 ‘일상생활 및 전공정보를 잘 다루기 위해’가 1,963명(55.6%)으로 과반수를 넘었다.



<프로그래밍 과목의 학습 이유>

학생들은 프로그래밍을 학습하여 전문 개발자가 되거나 혹은 컴퓨터과학 복수전공이나 전과보다는 자신의 분야에 융합하여 전공 지식을 보다 잘 활용하기 위함이었다. 특히 전공 과목을 본격적으로 배우기 시작하는 2학년 학생의 경우 이와 같은 요구가 다른 학년 보다 높았다.

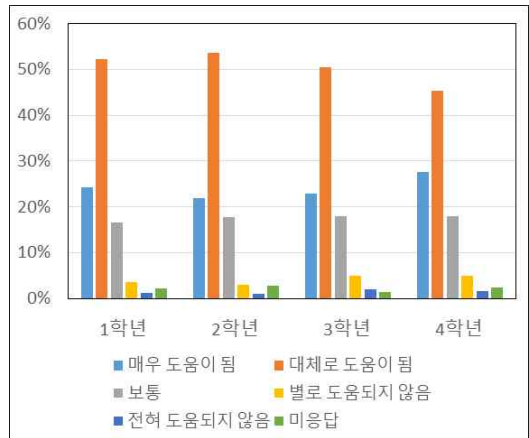
<프로그래밍 과목의 학습 이유에 대한 응답인원>

학년	인원	일상생활 및 전공정보를 잘 다루기 위해	취업	창업	기타	미응답	계
1학년	인원	612	297	105	70	34	1,118
	비율(%)	54.7	26.6	9.4	6.3	3.0	31.7
2학년	인원	475	184	88	49	2	798
	비율(%)	59.5	23.1	11.0	6.1	0.3	22.6
3학년	인원	425	190	79	54	0	748
	비율(%)	56.8	25.4	10.6	7.2	0.0	21.2

다음으로 취업에 유리한 요소가 될 수 있을 것이라 생각하는 학생이 906명(25.7%)이었으며 창업에 활용하기 위한 학생이 361명(10.2%)의 순이었다.

기타의 경우는 총 248명(7.0%)이었으며 이 중 취미 및 개인적 흥미(96명), 다방면에서 활용이 가능한 필수 능력이기 때문(12.1%), 전공이기 때문(7.3%) 등의 응답이 있었다.

다음으로 소프트웨어 교육이 본인에게 얼마나 도움이 될 것인가에 대한 조사를 하였다.



<소프트웨어 과목에 대한 유용성 기대치>

소프트웨어 교육이 본인에게 매우 도움이 되거나 혹은 대체로 도움이 될 것으로 긍정적인 응답한 학생은 2,639명(74.7%)로 매우 높은비율을 차지하였다. 반대로 소프트웨어 교육이 별로 도움되지 않거나 전혀 도움 되지 않는다는 응답은 198명으로 전체의 5.6%에 그쳤다. 앞서 언급한 첫 번째 문항에서 소프트웨어 과목을 수강하지 않을 계획인 학생이 350명(9.9%)이었음과 비교해보면, 소프트웨어 과목을 수강하지 않을 예정인 학생이더라도 그 필요성은 부분적으로 인식하고 있는 것으로 보인다.

설문 결과로 볼 때 학생들은 소프트웨어 교육에 큰 기대를 가지고 있으며 4차 산업 사회에서

프로그래밍과 같이 실용적인 기술을 학습하고 활용해야하는 필요성을 잘 인식하고 있었다.

<소프트웨어 과목에 대한 유용성
기대치에 대한 응답인원>

학년별	매우 도움이 됨	대체 로 도움이 됨	보통	별로 도움 되지 않음	전혀 도움 되지 않음	미응 답	계	
1 학년	인원	271	584	184	39	15	25	1,118
	비율(%)	24.2	52.2	16.5	3.5	1.3	2.2	31.7
2 학년	인원	174	427	142	25	8	22	798
	비율(%)	21.9	53.6	17.8	3.1	1.0	2.8	22.6
3 학년	인원	171	378	135	37	16	11	748
	비율(%)	22.9	50.5	18.0	4.90	2.1	1.5	21.2
4 학년	인원	240	394	155	43	15	21	868
	비율(%)	27.6	45.4	17.9	5.0	1.7	2.4	24.6
계	856	1,783	616	144	54	79	3,532	

2.2 소프트웨어 과목 설계 및 개설

앞 절에서 조사한 학습자들의 요구사항을 만족시키기 위해 소프트웨어 교과목을 개발하였다. 소프트웨어 교육은 컴퓨터과학 비전공자라면 모두 수강할 수 있어야 하므로 필수 교양의 형태로 2017년 1학기부터 과목을 개설 및 운영하였다.

개설 과목의 학습 목표는 컴퓨팅사고 이론 교육과 이를 통한 문제 해결 과정을 실제 소프트웨어로 구현할 수 있도록 프로그래밍 기술을 교육하는 것이다. 따라서 이론과 기술이 어느 한 쪽에 치우치지 않고 두 개념이 서로 밀접하게 연결된 과목의 구성이 필요하였다.

이에 맞게 과목의 구성은 주당 2시간의 이론 수업과 2시간의 실습 시간으로 구성하고, 이론 시간에는 컴퓨팅사고 개념과 해당 개념을 이용하여 문제를 해결할 수 있는 프로그래밍 언어 개념을 학습하고, 실습 시간에는 여러 가지 응용 문제들을 프로그래밍으로 해결하도록 하였다.

컴퓨팅사고 부분에서는 자료의 표현, 논리적 사고, 알고리즘 정의, 분할 전략, 모듈화와 같은 기본적인 문제 해결 개념과 문자열과 패턴 처리 등의 응용 기술까지 학습하도록 하였다.

프로그래밍 언어 학습 및 실습의 경우 데이터 입력, 연산, 그리고 출력의 기본 구조를 익히도록 하고, 너무 많은 라이브러리 함수나 사용 빈도가 낮은 세세한 명령문까지 학습하기 보다는 제어 구조의 작성에 집중하도록 하여 알고리즘적 사고 (algorithmic thinking)를 훈련하도록 하였다. 또한 학기말 프로젝트로는 팀 과제를 수행하게 하여 병렬 처리를 통한 문제 해결을 체험하도록 하였다. 전체적인 과목의 구성은 다음 표와 같다

<소프트웨어 교양 과목의 강의 구성>[8]

주차	이론		실습
	1교시(컴퓨팅사고)	2교시(프로그래밍)	
1	강의 소개		
2	컴퓨팅사고 개요	기본 입출력	1회
3	데이터 표현	변수와 데이터 유형	2회
4	논리	연산자와 수식	3회
5	알고리즘	제어 구조	4회
6	문제 해결 전략	함수	5회
7	중간시험		-
8	데이터 구조	리스트	6회
9	문제의 설계	리스트	7회
10	문자열 처리	문자열 처리	8회
11	패턴 처리	정규식 처리	9회
12	파일 처리	파일 처리	10회
13	병렬 처리	팀 기반 문제해결	팀 활동
14	심화 학습		-
15	자율학습		-
16	기말시험		-

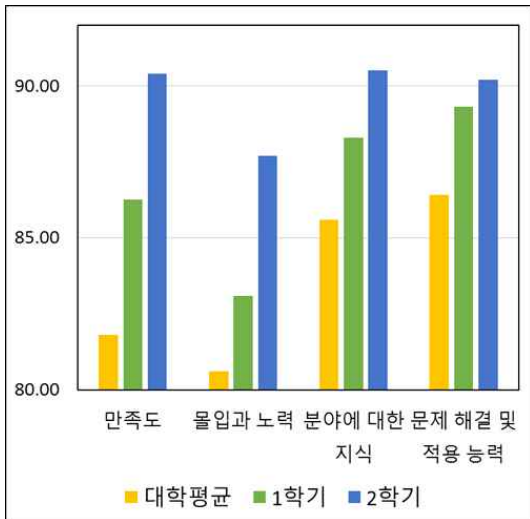
이와 같은 교양 과목 구성을 통하여 비전공자들이 기초 문제부터 복잡한 문제까지 컴퓨팅사고를 통한 해결 과정을 정의하고 이를 프로그래밍 언어로 구현 수 있도록 하고, 나아가 자신의 전공 분야에 적용할 수 있도록 하였다.

개발된 과목은 2017년 1학기과 2학기에 총 15개 분반, 각 40명 정원으로 개설되었으며 모든 분

반이 정원이 미달되지 않고 수강 신청이 이루어졌다.

2.3 과목의 평가 및 개선

각 학기말의 수강 평가 기간에는 수강생들로부터 만족도와 지식 향상 등의 6개 항목에 대한 평가를 받았다. 이 중 학습 효과와 직접적으로 관련이 없는 비차별 원칙 항목 등을 제외하고 만족도, 학생의 몰입과 노력, 해당 분야에 대한 지식, 그리고 문제 해결 및 적용 능력 항목에 대해 분석하였다.



<소프트웨어 과목에 대한 수강생의 피드백>

과목의 첫 개설 학기인 1학기에는 위 네 개의 항목에 대해 100점 만점 중 평균 86.7점의 값을 받았다. 대학 평균인 83.6점과 비교하면 학생들의 소프트웨어 과목에 대한 전체적인 만족도는 매우 높았다.

주관식 항목에서는 새로운 분야의 학문을 접하여 새롭고 프로그래밍이 흥미 있었다는 의견이 가장 많았다. 또한 수강 후 자신도 프로그래밍을 할 수 있게 되어 만족한다는 의견도 많았다. 부정적인 의견으로는 교양 과목임에도 학습량이 많고

프로그래밍 문제의 난이도가 높았다는 의견이 있었다.

이와 같은 의견과 한 학기 동안의 학생들의 제출물을 분석하여 집중적으로 실수를 하거나 오답을 작성하는 부분을 파악하여, 2학기 수업에서는 해당 내용을 더 자세히 설명하고 실습 문제를 보강하여 과목을 운영하였다. 그 결과 2학기에는 각 항목의 평균이 1학기 보다 향상되었다. 향후에도 컴퓨팅 문제 해결 역량을 키울 수 있도록 지속적인 과목의 개선 및 관리를 유지하는 것이 중요하다.

3. 결론

현재 우리나라의 소프트웨어 교육은 초기 단계로 해결해야 할 과제들이 많다. 소프트웨어 교육에 대한 높은 요구에 비해 인프라 구축, 교원의 확보, 그리고 교과 구성 등에 있어서 부족한 부분이 많으며 시급히 해결해 나가야 한다.

획일적인 교육 과정 구성이나 교육 시행 자체에 의미를 두어서는 안 되며 학습자, 교육기관 및 산업체 간의 요구사항을 명확히 파악하여 실용적인 교과를 구성해야 하며 학습자가 소프트웨어 기술을 활용하여 융합적 사고를 발휘할 수 있도록 교육해야 한다.

따라서 대학의 교양 과목으로서의 소프트웨어 교육은 단순한 코딩 교육이나 컴퓨터 개론과 같은 암기식 교육을 지양하고 컴퓨팅사고를 통한 문제 해결 능력과 기술을 훈련하여 다양한 문제에 적용이 가능하도록 기반을 마련해 주는 것이 중요하다.

마지막으로 초·중 교육과정에서 부터 소프트웨어 교육이 필수로 이루어지는 2018년 이후에는 각 교육 과정 간에 밀접한 논의를 통하여 체계적

인 교과를 구성하여야, 중복되는 내용을 방지하고 기초부터 고급 내용까지 단계적으로 학습할 수 있는 교육 과정을 구성할 수 있을 것이다.

참 고 문 헌

- [1] Klaus Schwab, The Global Competitiveness Report, World Economic Forum, 2016.
 - [2] www.gartner.com
 - [3] 교육부 고시 제2015-80호 (교육부, 2015)
 - [4] 초·중등 SW교육 필수화 준비 및 학교 중심의 SW교육 추진 (교육부·미래부, 2016)
 - [5] www.software.kr
 - [6] 학부대학 SW과목 개설에 대한 설문 (연세대학교 학부대학, 2016)
 - [7] yscec.yonsei.ac.kr
 - [8] portal.yonsei.ac.kr
-
-



김 재 경

- 2007년: 연세대학교, 컴퓨터과학과 박사
 - 2007년~2009년: University of Pittsburgh, Post-Doc
 - 2009년~2010년: 연세대학교, 연구교수
 - 2013년~2017년: 연세대학교, 객원교수
 - 2017년~현재 연세대학교 학부대학 조교수
 - 관심분야 : 소프트웨어 교육
-
-